Master of Science in Advanced Analytics

Calibrating High Frequency Trading Data to Agent Based Models using Approximate Bayesian Computation

Kelly Goosen

Supervisor: A/Prof. T. Gebbie

Mathematical Statistics,
Department of Statistical Sciences,
University of Cape Town, Rondebosch

2020

Abstract

We consider Sequential Monte Carlo Approximate Bayesian Computation (SMC ABC) as a method of calibration for the use of agent based models in market micro-structure. To date, there are no successful calibrations of agent based models to high frequency trading data. Here we test whether a more sophisticated calibration technique, SMC ABC, will achieve this feat on one of the leading agent based models in high frequency trading literature (the Preis-Golke-Paul-Schneider Agent Based Model (Preis et al., 2006)). We find that, although SMC ABC's naive approach of updating distributions can successfully calibrate simple toy models, such as autoregressive moving average models, it fails to calibrate this agent based model for high frequency trading. This may be for two key reasons, either the parameters of the model are not uniquely identifiable given the model output or the SMC ABC rejection mechanism results in information loss rendering parameters unidentifiable given insufficient summary statistics.

Keywords: agent based models, high frequency trading, calibration, approximate Bayesian computation, sequential Monte Carlo, stylised facts, market micro-structure.

Acknowledgements

Thank you to my supervisor, Tim Gebbie, for support, guidance and critique throughout this research and, moreover, for inspiring my interest in data driven finance research. As a lecturer and supervisor, he has taught me the core principles that have enabled this research as well as helped me present the work as clearly and thoroughly as possible.

Additionally, I would like to thank the Department of Statistical Science at the University of Cape Town for the solid grounding in statistics they provide for each of their students, the resources that have enabled me to complete this dissertation and for their continued support and advice during my studies.

I would also like to thank my peers for the happy and supportive atmosphere they created. In particular, thank you to Donovan Platt for his continued advice and valuable conversations that have expanded the breadth of my understanding and thank you to Michael Gant for his continued support and advice that enabled me to tackle the trickier technical challenges of this research.

Finally, I would like to thank my family and friends for their persistent love and encouragement throughout my studies.

Reproducing the Research

To reproduce the results please refer to the Python script files in the GitHub Repository Goosen and Gebbie (2020). Additionally, the code to reproduce figures and metrics can be also be found in the same GitHub repository. The data that was used to implement this research can be found in Gebbie and Platt (2019).

Contents

List of Figures

List of Tables

1 Introduction

We investigate whether Agent Based Models (ABMs) can be calibrated to High Frequency Trading (HFT) time series data. To date, there are no successful calibrations of agent based models to high frequency trading data (Platt and Gebbie, 2018). The validation of such models in literature has focused on stylised fact-centric model validation, where the model is assessed only by its ability to reproduce stylised facts (Platt, 2020). By stylised facts, we mean the nontrivial statistical properties that have persisted across a broad range of financial instruments and markets for more than 50 years (Staccioli and Napoletano, 2020). We provide a more robust validation by statistically optimising the parameters and use Bayesian credibility intervals to measure the degree of precision attained. We test the agent based model's ability to recover the original parameters with accuracy and a reasonably degree of precision using both synthetic data and Johannesburg Stock Exchange (JSE) trading data.

It serves as an extension to the work done by Platt and Gebbie (2018) who calibrated the Pries-Golke-Paul-Schnieder (PGPS) (Preis et al., 2006) liquidity-taker and -provider agent based model using the calibration technique: method of simulated moments (Platt and Gebbie, 2018). Platt and Gebbie (2018) concluded that either the method of simulated moments is not sufficient for calibrating a model as complex as the PGPS Model (Preis et al., 2006) to the market, or, that agent based models cannot be calibrated to market micro-structure data (Platt and Gebbie, 2018). According to Lux (2018), simpler calibration techniques, such as these, are insufficiently complex to estimate realistic intra-day time scale continuous double-auction markets. Accordingly, the next step in assessing whether HFT ABMs can be calibrated is to test a more flexible calibration technique that is able to incorporate more information from the time series HFT data.

As such, we implement Sequential Monte Carlo Approximate Bayesian Computation (SMC ABC) as an alternative more sophisticated method, recommend by Lux (2018). The results of this method are explored as an alternative to that of Platt and Gebbie (2018).

In Section 2, we compare and assess the merits of various HFT ABMs and the reasons behind selecting the PGPS Model (Preis et al., 2006). We then describe the PGPS Model (Preis et al., 2006) and its parameters in detail and the process of replicating and validating the model to ensure the model is correctly specified. Section 3 provides a deep dive into the calibration techniques that have previously been used in attempt to calibrate agent based models and reasons behind the selection of a Bayesian calibration approach. In Section 3.3.1.2 we provide a motivation for the sequential Monte Carlo approach to ABC and describe the SMC ABC algorithm in depth. We then detail the employment of the SMC ABC computation and the necessary hyper-parameters selected in Section 3.5 and Section 3.6, respectively. We assess and prove the validity of the implementation of SMC ABC by testing the calibration on a simple toy-model in Section 3.7. We then illustrate the poor calibration results on the synthetic high frequency data for the PGPS Model (Preis et al., 2006) in Section 4.1 and further illustrate in Section 4.2 that the calibrated model using JSE high frequency trading data fails to recover important stylised facts.

2 The Model

2.1 Agent Based Models

Agent based models enable the model to incorporate important traits that exist in reality in financial markets that are mostly ignored or incorrectly assumed in top-down economic modelling, namely that market participants make decisions using heuristics (rather than rationally), are heterogeneous (rather than identical), and interact in ways that result in feed-back loops (rather than independently) (LeBaron, 2001). Additionally, ABMs are particularly useful for modelling financial crisis events such as those researched

by Bookstaber et al. (2018) and Farmer and Geanakoplos (2009), with specific emphasis on pro-cyclical tail-risk events.

Zero-intelligence agent based models are models which do not rely on utility functions. They aggregate financial market trader decisions by stochastically sampling empirically generated distributions to determine agent behaviour (Farmer et al., 2005). ABMs, were first introduced in double auction markets by Gode and Sunder (1993) and they identified that market allocation efficiency arises due to market structure which is driven by trader motivation, intelligence and learning. Zero-intelligence ABMs have dominated ABM market micro-structure research and, promisingly, many zero-intelligence ABMs have been able to recover a number of order book dynamics, for example spread variance and price diffusion rates, most noticeably by Maslov (2000), Klimek et al. (2015) and, notably, by Preis et al. (2006) (Bookstaber et al., 2018).

2.2 The PGPS Agent Based Model

This research faces a similar issue to the primary problem facing tests of the efficient market hypothesis: the joint-hypotheses problem Malkiel (2003). The joint-hypothesis problem is where tests can fail either because one of the two hypotheses is false or because both parts of the joint hypothesis are false (Jensen, 1978). In the case of this research, there is a "triple-hypothesis" problem. Should we not be able to calibrate the ABM to the market, is it because the calibration technique is not sophisticated enough, is it because the ABM model is not appropriate for market micro-structure, or simply, are ABMs unsuitable for modelling market micro-structure? Therefore, it is of the utmost importance that an appropriate ABM is selected. The choice of the PGPS Model (Preis et al., 2006) is based on three desirable features and a consideration of the "triple-hypothesis" problem.

The first feature is the models ability to reproduce key stylised facts of the real market. It is a multi-agent–based order book model which is able to successfully reproduce the markets Hurst exponent for short, medium and long time scales and "fat tails" in the return distributions (Preis et al., 2006). Additionally, Platt and Gebbie (2018) have found that when calibrated parameters are used to initialize the PGPS Model (Preis et al., 2006), order-flow clustering is recovered, a key intra-day stylised fact of real financial markets (Platt and Gebbie, 2018). Secondly, the order matching algorithm while being fairly realistic is additionally fairly parsimonious. As such, the PGPS Model (Preis et al., 2006) is a significant shift conceptually away from closed-form approximations but still incorporates a reasonably large parameter space in comparison to simpler agent based models (Platt and Gebbie, 2018). The complexity of the model needs to be assessed carefully, The more complex the model, the more difficult it is to be calibrated and the slower the computational time. Thirdly, this ABM incorporates the zero-intelligence agent based model components which allows to assess whether incorporating more complex agent behaviours will improve the models ability to recover key stylised facts (Platt and Gebbie, 2018).

The PGPS Model (Preis et al., 2006) as it stands may not be able to recover important stylised facts. Optional additional complexity can be appended to the model by incorporating momentum traders (long position when prices have recently risen and short position when prices have decreased) which would likely lead to the models ability to reproduce volatility clustering (Mcgroarty et al., 2018). Additionally, one might consider building in a relationship between the price and volume of shares traded as one would expect the two factors to be highly correlated. This could be introduced using Copulas to identify the joint distribution or simply by identifying the correlation between the two factors in a uniform space. However, this is beyond the scope of this research and we continue with the existing PGPS Model (Preis et al., 2006) as it currently stands.

When we considering the "triple-hypothesis" problem, we need to refer to prior research. The latest investigation into whether ABMs can be calibrated to market micro-structure was investigated using the

PGPS Model (Preis et al., 2006). The research in this paper serves to extend the research provided by Platt and Gebbie (2018) and assess whether a more complex calibration technique can incorporate more information from the high frequency trading data to successfully calibrate the parameters of the PGPS Model. If a different ABM was selected, then it would not be possible to know whether a more complex calibration technique was required, as suggested by Lux (2018) or a different ABM was required. Using the same model alleviates this "triple-hypothesis" problem.

2.2.1 The Order Book

The PGPS Model (Preis et al., 2006) is an agent based model for simulating financial markets. The result is an order book for a single asset based on the trades of interacting agents in a continuous double auction virtual exchange. The outline for the model is provided in Section B.1 of the Appendix. The model assumes that the price development of a stock depends on the superposition of the actions of market participants and hence, macroscopic or emergent phenomena are derived from microscopic interaction. The emergent phenomena are the stylised facts attained by the model. Stylised facts are import features of real markets that are market- and time-independent properties. Emerging stylised facts from this model were the behaviour of the Hurst exponent in the short, medium and long time scales and the "fat tails" in the return distributions (Preis et al., 2006). The resulting simulated order book resembles the order book of a real exchange whereby market participants enter limit orders which provide liquidity to the order book or, conversely, enter market orders which remove orders from the limit order book. Limit order prices are set at or below the current best limit order price and are only executed if and when that price is the best limit order price and a market order is executed on the relevant side of the order book. Market orders, on the other hand, are executed immediately at the current best limit order price.

2.2.2 The Agents

According to the limit and market order, this order book model specifies two sets of interacting agents: the liquidity providers and the liquidity takers. There are N_A liquidity providers governed by a limit order execution rate of α per time-step, a default order size of one per agent and a $q_{provider}$ probability of a bid (or, equivalently, a $1 - q_{provider}$ probability of an ask). A bid and ask mean the price and quantity of a security that a trader is willing to buy and sell, respectively. The number of liquidity takers is set to equal the number of liquidity providers (N_A) but the market order execution rate of liquidity takers is given by μ per time-step, a default order size of one per agent and is a buy and sell market order with probability q_{taker} and $1 - q_{taker}$, respectively. Additionally liquidity takers can cancel their limit orders with a probability of δ per time-step, so that limit orders can be removed by market orders or by being cancelled. Therefore, in each time-step, approximately floor$(N_A\alpha)$ limit orders are placed, floor$(N_A\mu)$ market orders hit the order book removing limit orders and (Number of Limit Orders $\times\ \delta$) are cancelled. Consquently, we require $\alpha(1-\delta) > \mu$ and $\delta > 0$ for a stable order book, the derivation of which is provided in the Part D.1 of the Appendix. Through this order matching algorithm the model attains a price time priority which is usually exhibited in real markets (Mandelbrot, 1967). The limit bid order probability is fixed at $q_{provider} = \frac{1}{2}$. Whereas, the market order buy probability is time varying and is specified by a mean-reverting random walk with mean $q^0_{taker} = \frac{1}{2}$, increment size $\pm\Delta_s$ each time-step and mean reversion probability $\frac{1}{2} + |\, q_{taker} - \frac{1}{2}\,|$. This specification results in a Hurst exponent $> \frac{1}{2}$ on intermediate time scales which is realistic in financial markets as the temporary trends result in non-stationary price increment behaviour (Preis et al., 2006).

The price of the liquidity provider limit order depends on the current time-step (t), best ask ($p_a(t)$) and best bid ($p_b(t)$) and is given by:

$$\text{Limit buy order price} = p_a(t) - 1 - \nu(\lambda(t), u) \tag{1}$$

$$\text{Limit sell order price} = p_b(t) + 1 - \nu(\lambda(t), u), \tag{2}$$

where:

$$\nu(\lambda(t)) = \text{floor}(-\lambda(t)\log(u)). \tag{3}$$

Here, $u \sim U(0,1)$ a Uniform random number and $\lambda(t)$ is a time dependent order placement depth parameter which depends both on the time-step and the buy probability of liquidity takers and is calculated as follows:

$$\lambda(t) = \lambda_0 \left(1 + \frac{|\, q_{taker}(t) - \frac{1}{2}\,|}{\sqrt{(q_{taker}(t) - \frac{1}{2})^2}} C_\lambda\right). \tag{4}$$

This non-static entry depth parameter results in the emergence of "fat-tails", whereas, when $C_\lambda = 0$ the return distributions are Gaussian. The depth parameter factors in two important features, the relative strength of the current trend and the historic volatility. The denominator in the fraction signals the strength of the current market order bid probability which is standardized by dividing by the historic volatility, thereby indicating if the there is currently a weak or strong trend relative to the historic trajectory of the asset (Preis et al., 2006). In trend-less markets, large price movements are not expected and, hence, liquidity providers will place bids close to the midpoint to exploit small price fluctuations. Conversely, when trends are strong, liquidity provider risk increases for orders placed close to the midpoint and hence, to decrease market risk, liquidity providers lengthen their order placement depth. The numerator is calculated separately before the main simulation starts for 10^5 Monte Carlo steps as the process safely converges to its true standard deviation for this number of steps which is shown in Figure 1. This is shown in the flow diagram provided in Section B.1 where the variance is computed in the initialisation phase of the model before the simulated trading begins.

Following the model specifications detailed above, two theoretically independent stylised facts are attained, the non-stationary behaviour of the price increments and the "fat-tails" of the return distributions. Both stylised facts emerge as a results of the market-maker agent specification of the mean reverting random walk of the market buy probability, $q_{taker}(t)$, and the response of the liquidity providers to adjust for market risk during trends (Preis et al., 2006).

2.3 The PGPS Model Replication

The PGPS Model (Preis et al., 2006) is reprogrammed in Python and can be found in the GitHub repository Goosen and Gebbie (2020).

To ensure the PGPS Model (Preis et al., 2006) construction is done correctly and without error we needed to ensure that the resulting order book and mid-price path generated through our construction resulted in the exact same results as that given by Platt and Gebbie (2018) (given the same parameter inputs). To replicate these results, an investigation was required into three core components: the model construction, the model output and the pseudo-random numbers incorporated.

Originally, Platt and Gebbie (2018) constructed the PGPS Model (Preis et al., 2006) in Matlab using a heavily object oriented approach (MATLAB, 2010). As such, a similar objected oriented program is selected to re-construct the PGPS Model (Preis et al., 2006): Python (Van Rossum and Drake, 2009). To replicate the model, the same classes used by Platt and Gebbie (2018) were used in the Python implementation. This made it easy to spot and remove any differences from the Python code to the underlying Matlab construction as well as validate that the original implementation is done correctly.

After the model was constructed, the next step was to run the code in both applications using the same parameters to ensure the model outputs were identical. Platt and Gebbie (2018)'s Matlab PGPS Model (Preis et al., 2006) was run using Octave GNU to produce a set of model outputs (John W. Eaton and

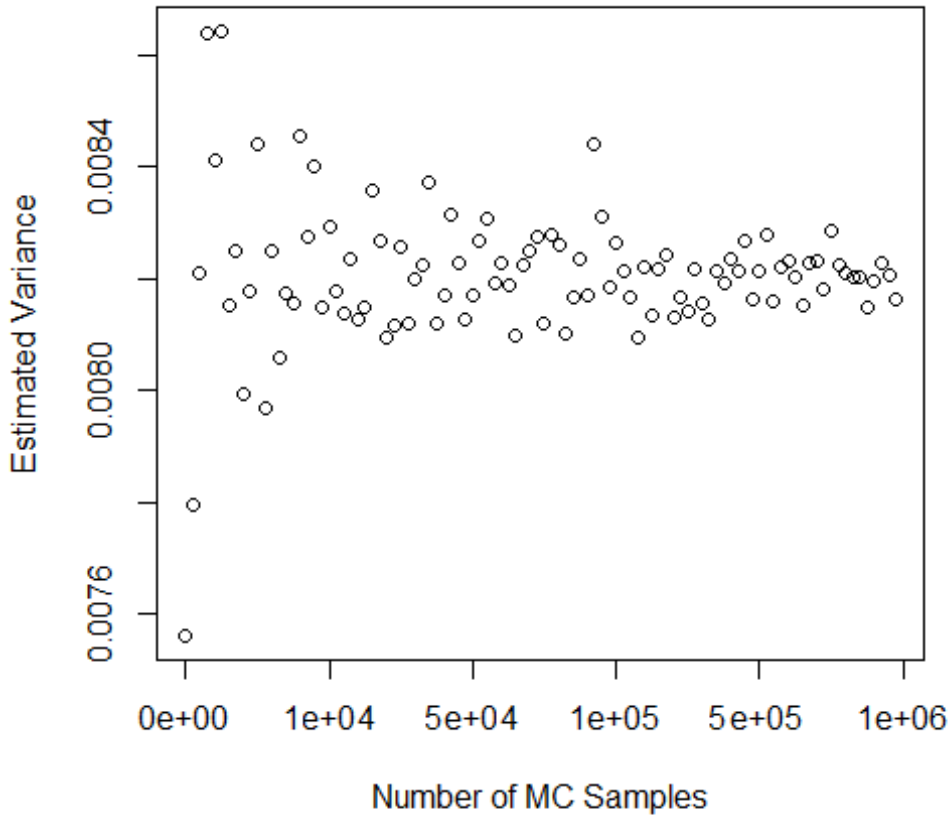

Figure 1: Plot showing the number of Monte Carlo simulations required to estimate the variance of the price movements. Consequently, the use of 10^5 Monte Carlo simulations is sufficient to estimate this underlying volatility in the price as convergence is achieved after approximately $50,000$ simulations.

Wehbring, 2014). However, even after setting the seed and using the same random generator (Merscene Twister) for both the Matlab code and Python code, it was identified that that the pseudo-random numbers generated were not identical across coding platforms after more than 100 numbers generated (Matsumoto and Nishimura, 1998). A number of varying pseudo-random number generators were investigated but all of which had the same results: a disconnect between the random number generators across coding platforms Matlab and Python.

As such, a work-around was required to ensure we had exactly and correctly constructed the PGPS Model (Preis et al., 2006). In Matlab we simply ran the PGPS Model (Preis et al., 2006) using the original code provided by Platt and Gebbie (2018) but used the Merscene Twister pseudo-random number generator and specifically set the seed (see the Matlab construction provided in GitHub Repository Goosen and Gebbie (2020)) (Matsumoto and Nishimura, 1998). Additionally, in Matlab we used the same pseudo-random number generator and identical seed to simulate one million random numbers - meaning that this set of random numbers aligns exactly with the numbers used in the Matlab implementation of the model. These numbers were saved and, subsequently, read into our Python construction of the PGPS Model (Preis et al., 2006). This ensured that an identical price path and order book could be constructed in our Python implementation of the PGPS Model (Preis et al., 2006) as that of the original Matlab construction provided by Platt and Gebbie (2018).

After a few alterations of our Python construction of the PGPS Model (Preis et al., 2006) and using the default parameters given Table 1, we were able to simulate a price path (as shown in Figure 3) that is identical to that of price path simulated using the same parameters under Platt and Gebbie (2018) (shown in Figure 2). Thus, proving the model has been constructed and implemented correctly.

Table 1: Default model parameter set presented by Preis et al. (2006) and used in reproducing simulated intra-day price path in Python and Matlab (Octave GNU).

Parameter	Default Value
δ	0.0250
λ_0	100
C_λ	10
Δ_S	0.0010
α	0.1500
μ	0.0250
N_A	125

3 Calibration

Despite the recent paradigm shift towards ABMs, there is criticism on the inadequacy of current ABM validation approaches (Grazzini and Richiardi, 2015). Qualitative validation, where the model is assessed only by its ability to produce empirically-observed stylised facts, is as far as most studies go to in terms of assessing model validity. This is particularly the case for more complex ABMs with larger parameter spaces (Panayi et al., 2012) (Guerini and Moneta, 2017). Although this stylised fact-centric model validation appears reasonable at first, it fails on two accounts. The first account is due to the adhoc manner parameters are estimated in these studies to produce the stylised facts. The more parameters in the model, the greater the parameter space and therefore, the more combinations of parameters that can be specified to suitably "fit" the desired stylised facts. We only need to look at top-down models and the purpose of dimension reduction techniques (such as principle components analysis) to understand why this technique is flawed (Kambhatla and Leen, 1997). Consequently, in misspecifying these parameters, one loses a primary advantage of ABMs, the ability to perform sensitivity analysis around these parameters. By

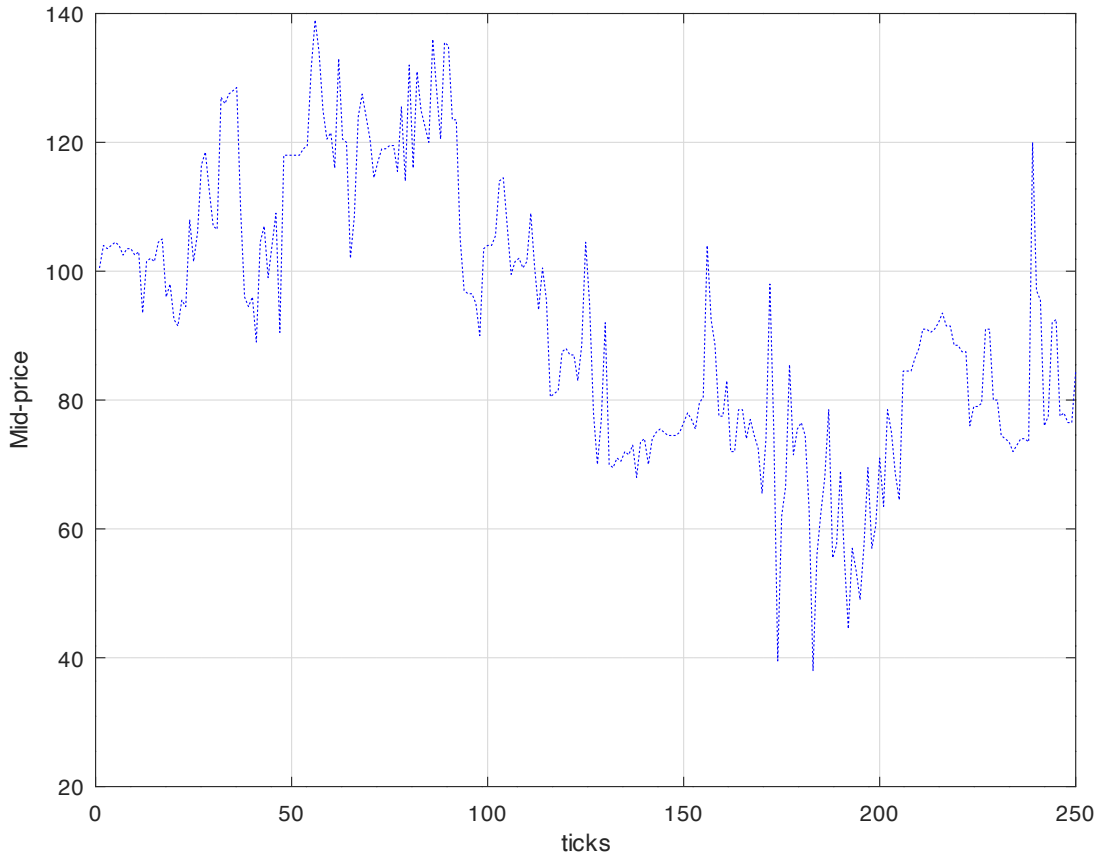

Figure 2: Plot showing the simulated mid-price path for $T = 250$ time-steps using Platt and Gebbie (2018) original code formulation of the PGPS Model (Preis et al., 2006). The simulation and plotting is constructed in Octave GNU using the original Matlab code. The Merscene Twister pseudo-random number generator in Octave is used with a seed of 1 (Matsumoto and Nishimura, 1998). The intra-day price is the mid-price between the best ask and best bid in the limit order book at each time-step. The parameters used are given in Table 1.

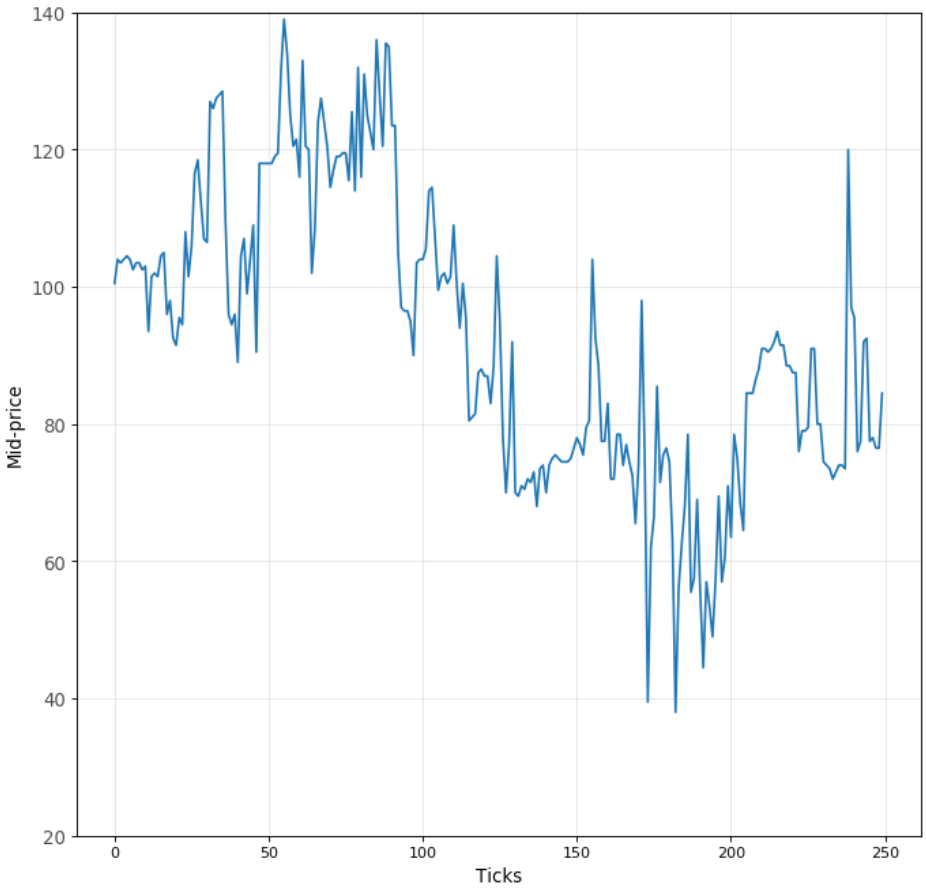

Figure 3: Plot showing the simulated mid-price path for $T = 250$ time-steps produced in Python. The simulation reproduces the exact price path and limit order book as the original Matlab code (Platt and Gebbie, 2018). The Merscene Twister pseudo-random number generator in Octave is used with a seed of 1 and the random numbers are transferred to Python to replicate the intra-day price path (Matsumoto and Nishimura, 1998). The intra-day price is the mid-price between the best ask and best bid in the limit order book at each time-step. The parameters used are given in Table 1.

altering the parameters from their calibrated estimates, one can not only test the robustness of the model but, additionally, one can test hypotheses around how changing the underlying ABM parameters might alter the emergent dynamics of the system. The second issue with qualitative stylised fact centric model validation is the large number and wide variety of models that recover a similar number of these empirically observed stylised facts but have vastly different behavioral rules. Due to this, there is no room to infer causal effects from these models and, moreover, results in robust model comparison a practical impossibility (LeBaron, 2006), (Barde, 2016), (Lamperti, 2015)).
Consequently, arising from these criticisms is a small but growing number of studies investigating sophisticated quantitative calibration approaches. Unfortunately, the research is compartmentalised in that researchers introduce new methods without comparing them to former alternatives and, additionally, for majority of these techniques, the theoretical properties of the estimates are not well understood (Grazzini and Richiardi, 2015). Moreover, the techniques are only applied to simple models with few parameters. As a result, there are a large number of calibration techniques with little indication as to which technique is preferred particularly for larger and more complex models.

3.1 Method of Simulated Moments

Fagiolo (2016) has produced an extensive investigation into the calibration of agent based models. In the majority of this research, simulated method of moments has been used as a calibration technique. Additionally, information theoretic criteria presented by Lamperti (2015) and Barde (2016), Bayesian estimation presented by Grazzini et al. (2017) and surrogate modelling presented by Lamperti and Sani (2017) has been investigated. However, this research has primarily focused on calibration of simple agent based models with closed form approximations on inter-day market prices. In particular, Farmer and Joshi (2002) and Brock and Hommes (1998) have calibrated models of this nature.

A branch of more complex agent based models have been developed for intra-day modelling of continuous double auction markets. Models at the forefront of this research are given by Preis et al. (2006), Chiarella and Iori (2002), and Chiarella et al. (2009) but calibration of the above models through robust numerical calibration is largely unexplored research (Platt and Gebbie, 2018). Platt and Gebbie (2018) have calibrated the PGPS Model (Preis et al., 2006) using simulated method of moments. This use of this calibration technique was driven by a number of considerations. Firstly, it is selected for its transparent and simple implementation of the method of simulated moments, which has gained prominence in calibration literature (Platt and Gebbie, 2018). Secondly, the method of simulated moments selects moments based on their relevance to stylised facts meaning calibrated models using this approach are more likely to recover stylised facts. Subsequently, following financial agent based model research which has focused primarily on stylised fact validation, the method of simulated moments is a reasonable successor. Thirdly, consistency and uncertainty around the method of simulated moments estimators are broadly understood in literature. This is an advantage over modern calibration approaches, the properties of which are not yet entirely understood (Platt and Gebbie, 2018).

However, the method of simulated moments is constrained in the way it summarises the data into a finite number of moments (Platt and Gebbie, 2018). In this way a large amount of information is lost because the moments are not sufficient.

3.2 Bayesian over Frequentist Calibration

There are two broad approaches to calibrating economic agent based models in literature, Bayesian estimation and objective function-based Frequentist approaches. Platt (2020) overcomes the aforementioned obstacles of choosing between the two approaches by considering a wide variety of both Bayesian and Frequentist calibration techniques against a number of varyingly complex models. The findings conclude

that, although Bayesian is less popular in literature, it consistently outperforms Frequentist techniques and produces parameter estimates that are reasonable in many contexts.

Additionally, Frequentist methods, such as the method of simulated moments described above, generates a point-estimate rather than a posterior distribution. Without the posterior distribution, Frequentist methods cannot be used to to calculate various point estimates. This is relevant as each point estimate (the mean, median and mode), can be used to minimise different loss functions (the squared error loss, absolute value loss and the 0-1 loss, respectively) (Platt, 2020).

3.3 Approximate Bayesian Computation (ABC)

More recently, calibration termed Approximate Bayesian Computation (ABC) has been at the forefront of Bayesian inference. ABC provides a way of evaluating posterior distributions of analytically or computationally intractable likelihood functions (Sisson et al., 2007). There has been a vast and varied uptake of these models which span ecology (Butler and Glasbey, 2009), molecular genetics (Marjoram and Tavaré, 2006), epidemiology (Tanaka et al., 2006) and extreme value theory (Bortot et al., 2007). By posterior distribution we mean, given both the prior distribution of the parameters, $\pi(\theta)$, and the likelihood function $f(x \mid \theta)$, the posterior distribution is given by:

$$f(\theta \mid x) \propto \pi(\theta) f(x \mid \theta) \tag{5}$$

i.e. the probability of the parameters given the observed data, which we want to maximize.

However, as a work around for intractable likelihood functions, all ABC algorithms follow the generic procedure given in Algorithm 1.

Algorithm 1 ABC Algorithm Overview

Require:

1. A candidate parameter (set) is generated, $\theta*$.
2. Given (conditioning on) the candidate parameter, $\theta*$, and the likelihood function, $f(y \mid \theta*)$, a simulated data set, $y*$, is generated.
3. The candidate is accepted if the simulated data, $y*$, and the observed data, y_{obs}, are sufficiently "similar".

Similarity, in Algorithm 1, is measured by a distance function, d, (e.g. Euclidean distance) between the summary statistic vector, $S(y)$, of the observed and simulated data and an acceptance tolerance of the distance given by ϵ. So that we accept the candidate if and only if:

$$d(S(y_{obs}), S(y*)) \leq \epsilon. \tag{6}$$

As such, by following the ABC algorithm, we are sampling from the joint distribution:

$$f(\theta, y \mid d(S(y), S(y*)) \leq \epsilon). \tag{7}$$

Ultimately, we are interested in inferring the optimal value of θ and consequently, are interested in the marginal distribution:

$$f(\theta \mid d(S(y), S(y*)) \leq \epsilon). \tag{8}$$

Finally, if the sufficient statistics, $S(y)$, are near sufficient and the tolerance ϵ, is reasonably small, then $f(\theta \mid d(S(y), S(y*)) \leq \epsilon)$ should approximate $f(\theta \mid y)$.

3.3.1 Computational Techniques for ABC

3.3.1.1 Simple Rejection ABC

A number of computational techniques have been developed to improve the efficiency of ABC inference. These methods adapt simple rejection ABC, given in Algorithm C.1, to use smaller values of ϵ. Since convergence is achieved as $\epsilon \to 0$, but computational time increases as ϵ decreases because more simulations are rejected. Simple ABC performs particularly poorly when the data are informative resulting in a posterior distribution that has a more concentrated density than the prior distribution (Beaumont et al., 2009). The first general approach to improving the efficiency of the proposal distribution is to alter the proposal distribution, for example using regression correction methods at the outset (Fearnhead and Prangle, 2012). Alternatively, what is applied more widely are two methods, Markov Chain Monte Carlo (MCMC) and sequential Monte Carlo ABC, which sequentially alter the proposal distribution.

3.3.1.2 Sequential Monte Carlo (SMC) ABC

However, SMC methods have been brought to prominence over MCMCs due to the difficulties of designing efficient MCMCs (Moral et al., 2011). In particular, SMC ABC acceptance kernel permits a smaller bandwidth and proposes a more efficient proposal distribution to that of MCMC (Beaumont et al., 2009). SMC ABC methods successively repeat MCMC calibrations (described in Algorithm C.2 in the Appendix) and with the t^{th} repetition generate new proposal distributions from which to sample, $q_t(\theta)$, from the previous step. In that way, we sample from $q_{t-1}(\theta)$ instead of the prior distribution, $\pi(\theta)$, in the t^{th} MCMC iteration. This is visually illustrated in the flow diagram in Section 19 of the Appendix. An approximate kernel density, $K_t(\cdot)$, is used to approximate the density function. The initial $q_1(\theta)$ is often taken as $\pi(\theta)$ and subsequent proposal distributions have the following density function:

$$q_t(\theta) = \left(\sum_{i=1}^{N} w_i^{t-1} K_t(\theta \mid \theta_i^{t-1}) \right) / \sum_{i=1}^{N} w_i^{t-1}. \tag{9}$$

Beaumont et al. (2009), Peters et al. (1993), Sisson et al. (2007) and Toni et al. (2009) have made advances in a branch of SMC ABC termed Population Monte Carlo (PMC) ABC and the most up-to-date version of which is given in Algorithm 6 in the Appendix. However, in 2011, Moral et al. (2011) developed an adaptive SMC ABC which is preferred for two reasons over PMC ABC. The first is that the adaptive SMC ABC's computational complexity is linear in the number of simulations, whereas, PMC ABC's is quadratic. This is important when simulating computationally intensive models, like the PGPS Model (Preis et al., 2006). The second is that both the sequence of tolerance levels and the parameters of the proposal distribution are automatically generated for SMC ABC. On the other hand, for PMC ABC the tolerance levels need to be carefully constructed since, if the tolerance levels decrease to quickly, the algorithm performs poorly and if too slowly, the computational time becomes to burdensome (Moral et al., 2011).

SMC ABC Algorithm The SMC ABC is an adaptation of the PMC ABC algorithm (Algorithm 6) which, instead, generates particles using a Metropolis Hasting proposal kernel (as in MCMC Algorithm C.2) and additionally, re-samples the particle weights (bootstrapping the particle filter) (Beaumont et al., 2009). This allows the algorithm to automatically update the tolerance levels, ϵ^t, with the aim of accepting a proportion, ν, of particles produced by the tolerance in ϵ^{t-1}, so that:

$$\nu \sum_{i=1}^{N} K_{\epsilon_{t-1}}^{r} \left(d(s^*, s_{obs}) \right) = \sum_{i=1}^{N} K_{\epsilon_t}^{r} \left(d(s^*, s_{obs}) \right). \tag{10}$$

The initial tolerance is set so that $\epsilon^0 = \infty$ and the ϵ^1 needs to be specified by the user, which should be set suitably large by empirical exploration. Gallant et al. (2018) and Blevins (2016) demonstrate that the

particle filter approximation is unbiased and symptomatically consistent, respectively. The Moral et al. (2011) SMC ABC algorithm then proceeds as described in Algorithm 2.

Algorithm 2 SMC ABC

Require:

1. At $t = 0$, for $i = 1, 2, 3, ..., B$:
   1.1. Sample candidate parameters from prior, $\theta_i^* \sim \pi(\theta)$.
   1.2. Given candidate parameter, θ_i^*, simulate $s_i^* = S(f(y_i^* \mid \theta_i^*))$.
   1.3. Let $w_i = 1/B$.
2. Set $t = t + 1$, and:
   2.1. Compute ϵ_t using ν and ϵ_{t-1} as given in Equation 10.
   2.2. For particles $i = 1, 2, ..., B$ which are accepted, set $w_i = 0$.
   2.3. Normalize the weights such that $\sum_{i=1}^{B} w_i = 1$.
   2.4. Compute the effective sample size: $ESS = \left(\sum_{i=1}^{B} w_i^2 \right)^{-1}$.
   2.5. If $ESS < B/2$, re-sample B particles with probability w_i.
   2.6. For all particles with $w_i > 0$, use Metropolis Hastings perturbation kernel to perturb the particles (i.e. repeat Steps 2-4 from MCMC ABC Algorithm (Algorithm C.2).
3. Continue until stopping criteria is reached, one of:
   3.1. $\epsilon_t \leq \epsilon^{(min)}$,
   3.2. Minimum Particle Acceptance Rate, $PAR_t \leq PAR^{(min)}$, or
   3.3. Number of populations, $t \geq t^{(max)}$,

   where $\epsilon^{(min)}, PAR_t^{(min)}$ and $t^{(max)}$ are specified by the user through empirical exploration of the calibration results.

3.4 Computational Complexity

The computational complexity of the PGPS Model (Preis et al., 2006) means that simulating a price path of 2300 minute bars enough times to numerically calibrate parameter estimates is practically infeasible using even the most advanced 8 core processor. A number of steps were taken to speed up the process. The first of which is by using built in functions and packages since they are efficient and well-tested. The second is vectorizing the code and avoiding unnecessary loops. The third is ensuring that computations were performed using facilities provided by the University of Cape Town's ICTS High Performance Computing (UCT HPC) team: hpc.uct.ac.za - which meant that the computations could be performed across 40 cores simultaneously. Finally, profiling of the PGPS Model (Preis et al., 2006) code was done to identify which steps of the model were slowing down the implementation the most and these steps were optimised to reduce the compute time of the model. Additionally, unit tests were used to ensure any optimisations to the code do not change the underlying model output (see tests in GitHub repository Goosen and Gebbie (2020)). Considering the speed is essential for calibrating the model as, the underlying PGPS Model (Preis et al., 2006) is fairly complex and computationally intensive and most numerical calibration techniques require thousands of simulations of the underlying model in order to estimate the parameters. For example, the SMC ABC calibration of the model using a price path of 2300 Monte Carlo steps takes 3 days, 12 hours, 45 minutes to run on the facilities provided by the University of Cape Town's ICTS high performance computers using 40 cores in parallel. We found that overall the success of the calibration of complex models is heavily constrained by the computational complexity of both the underlying model and the implementation of the calibration technique.

3.5 Calibration using pyABC

PyABC is a scientific library written in Python for distributed, likelihood-free inference using Approximate Bayesian Computation (ABC) (Klinger, 2018). In particular, pyABC is used for approximate Bayesian computation using the sequential Monte Carlo implementation, which is a particularly efficient ABC algorithm. We use SMC ABC to understand the posterior distribution of the underlying model parameters. In this case, we use pyABC to understand which parameters explain the underlying data. If the parameters can be calibrated then high frequency trading data can, at least partly, be explained using agent based models. If we were able to analytically write down the likelihood function for the PGPS Model (Preis et al., 2006) then using ABC would not be an appropriate approach. This package should only be used for the much harder likelihood-free inference, like the calibration problem we have here.

The package pyABC is particularly useful in this case because the PGPS Model (Preis et al., 2006) as well as the SMC ABC algorithm are computationally expensive and pyABC is specifically designed to run efficiently on multi-core machines and distributed cluster setups (Klinger, 2018). This library offers two different parallelisation strategies: dynamic scheduling and static scheduling which have their respective advantages and drawbacks. Dynamic scheduling is used in our case as it is notably more scalable than static scheduling when the number of cores available is large (Klinger, 2018). Using this library on the University of Cape Town's high performance computers we are able to implement a distributed run of sequential Monte Carlo approximate Bayesian computation on forty cores. One disadvantage of this package for those looking to run SMC ABC through multi-core and distributed setups is that this particular functionality is not available for Microsoft Windows. However, pyABC is currently the most efficient ABC framework given distributed infrastructure. Due to the complexity of both ABC and the underlying model, calibration of parameters would be impossible if not for a highly efficient framework (Klinger, 2018).

Importantly, pyABC adheres to software engineering practices and established design patterns (Klinger, 2018). PyABC allows for a rich set of default choices, and is highly customisable. Post-processing of pyABC results is facilitated through pyABC's Application Programming Interface (API) for data querying. Logging and storage of SMC ABC runs are easily implemented and the library is well documented incorporating a user guide, developer guide and Jupyter Notebook tutorial examples. For these reasons pyABC has been used by several research groups worldwide, notably pyABC calibrated parameters for stochastic models of HIV spread (Imle et al., 2019).

3.6 SMC ABC Customisable Features and Hyper-Parameters

3.6.1 Stochastic Simulator

To use approximate Bayesian computation, we need a parametrized stochastic simulator to numerically generate samples from the specified model. The PGPS Model (Preis et al., 2006) is the relevant ABM of choice and we use the Python implementation described in the Section 2.2 to numerically draw samples of data given the underlying model parameter inputs.

3.6.2 Synthetic Data

Additionally, we need observed or synthetically generated data to which the underlying parameters are calibrated. Initially we synthetically generate data. The reason for this is twofold: firstly, we generate synthetic data to ensure the underlying PGPS Model (Preis et al., 2006) is identical to that of Platt and Gebbie (2018); secondly, the smaller set of synthetic data is used to tune the ABC hyper-paramters to ensure that the ABC is as efficient as possible for this computationally intensive algorithm. Both the underlying PGPS Model (Preis et al., 2006) and the ABC implementations are computationally expensive so the initial synthetic data has a price path time horizon of 100.

Table 2: Parameter range (minimum and maximum bound) defined for each of the six parameters around the default set presented by Preis et al. (2006) and used for simulating the parameters according to a Uniform distribution with the lower and upper bounds as given above..

Parameter	Minimum	Maximum
δ	0.00	0.050
λ_0	50.00	300.000
C_λ	1.00	50.000
Δ_S	0.00	0.005
α	0.05	0.500
μ	0.00	0.050

Table 3: Default summary statistic set presented by Platt and Gebbie (2018) used to summarise the price path.

Summary Statistic	Description
$\bar{Y} = \sum_{t=0}^{N} Y_t$	Sample Mean
$\sigma_y = \frac{1}{N}\sum_{t=0}^{N}(Y_t - \bar{Y})^2$	Sample Standard Deviation
$\text{skew}_y = \frac{1}{N}\sum_{t=1}^{N}\left(\frac{Y_t-\bar{Y}}{\sqrt{\sigma_y}}\right)^3$	Skewness
$k_y = \frac{1}{N}\sum_{t=1}^{N}\left(\frac{Y_t-\bar{Y}}{\sqrt{\sigma_y}}\right)^4 - 3$	Kurtosis
$KS_y = \max\{F(y) - G(y_{obs}) : y = 1, 2, 3...\}$	Kolmogorov Smirnov Test Statistic
Regress: $\log\left(\frac{R}{\sigma_y}\right)_n = \log(c) + H_y \log(n)$	Hurst exponent,H_y

3.6.3 Model Parameters

The model parameters which we want to calibrate need to be inputted into the SMC ABC calibration algorithm along with their prior distributions. We use the pyABC default non-informative prior, independent uniform distributions for each of the model parameters over the ranges specified in Table 2.

3.6.4 Output Function

The output function is broken up into two parts. The first is the model and the second is the summary statistics which summarise the output from the model. The model in this case is the PGPS Model (Preis et al., 2006) which simulates price paths given the parameters generated from the prior distributions. The summary statistic function summarises the price path into a finite number of metrics. We trial two sets of summary statistics. The first are the summary statistics specified by Platt and Gebbie (2018) which are listed in Table 3 and the second uses the same six summary statistics but incorporates five auto-correlation lag statistics.

In Table 3, the kurtosis measures how fat- or thin-tailed the distribution of returns is relative to a Normal distribution. Fat tails in return distributions have been observed across all time scales and in many markets, including Euronext, the LSE, NASDAQ, American Stock Exchange and Shenzhen Stock Exchange (Plerou and Stanley, 2008) (Cont, 2001) (Chakraborti et al., 2011).

In the Kolmogrov Smirnov (KS) - statistic, $F(y)$ and $G(y_{obs})$ are the observed cumulative distribution functions of the simulated and observed price paths, respectively. This is a two-sided test for the hypothesis that the two independent samples are drawn from the same continuous distribution (Hodges, 1958).

Algorithm 3 Hurst exponent

The Hurst exponent subdivides the price path of length N into d sub series $(Y_{s,r})$ of length n (Granero et al., 2008). Using this formulation, the Hurst exponent is computed as follows:

Require:

1. Find the mean, $\bar{Y}_r$, and standard deviation, σ_r, of each sub series $Y_{s,r}$.
2. Normalise $Y_{s,r}$, i.e. $X_{s,r} = Y_{s,r} - \bar{Y}_r$.
3. Calculate the cumulative time series $Z_{s,r} = \sum_{j=1}^{s} X_{j,r}\ \forall s = 1, 2, ..., n$.
4. Calculate the range $R_r = \max\{Y_{s,r} : s = 1, ..., n;\ r = 1, ..., d\} - \min\{Y_{s,r} : s = 1, ..., n;\ r = 1, ..., d\}$.
5. Re-scale the range R_r/σ_r.
6. Find the mean value, $(R/\sigma)_n$, of the re-scaled range for all d sub-series of length n.

Based on Algorithm 3 and the fact that the statistic R/S is asymptotically related to the Hurst exponent as follows:

$$(R/S)_n \approx cn^H, \tag{11}$$

we can use linear regression to estimate H. The Hurst exponent is used to detect long memory in time series. Note that H necessarily lies between 0 and 1 and has the following properties depending on the value it takes on: If $H = 0$, the series is a white noise, if $H > 0.5$ or $H < 0.5$ then the time series is persistent and anti-persistent respectively. This specification results in a Hurst exponent $> \frac{1}{2}$ on intermediate time scales which is realistic in financial markets as the temporary trends result in non-stationary price increment behaviour (Preis et al., 2006). The process is a Brownian motion if $H = 0.5$ and is a simple linear trend if $H = 1$ (Granero et al., 2008). The Hurst exponent is used to measure volatility clustering, the long memory of square or absolute value of returns which results in large price changes following other large price changes. Volatility clustering has been found to persist across a financial markets, including NYSE and S&P 500 index futures (McGroarty et al., 2019). Emperically, research as shown that the Hurst exponent varied from a minimum of $H \approx 0.58$ on the Shenzhen Stock Exchange to a maximum of $H \approx 0.815$ for the USD/JPY currency exchange (McGroarty et al., 2019). Volatility clustering is prevalent due to two primary causes, timed-execution of large orders and the arrival of new information (Bouchaud et al., 2004).

The advantage of using SMC ABC over simpler methods such as method of moments used by Platt and Gebbie (2018) is that we are able to incorporate more information to improve the calibration. As such, for the second set of summary statistics we use the six default statistics and, additionally, incorporate auto-correlation statistics up to 5 lags. The Auto-Correlation Function, $\text{ACF}(\gamma) = \text{corr}(Y_t, Y_{t+\gamma})$ for a given lag indicates the degree of similarity between a given time series and its lagged series. Across a number of varied markets, including FX markets, the NYSE, S&P 500 Index and Euronext, price time series do not possess significant auto-correlation with the exception of weak negative auto-correlation at short time scales (Aït-Sahalia et al., 2011) (Chakraborti et al., 2011). Intuitively this can be explained through a no-arbitrage argument because if there were significant auto-correlations then traders would implement strategies to exploit the auto-correlation and generate risk-free trading profit. Subsequently, these strategies would reduce the auto-correlation such that it no longer exists. Statically, in recent years the weak negative auto-correlation has eroded more quickly which is likely a result of increasingly more efficient financial systems (McGroarty et al., 2019).

3.6.5 Hyper-parameters

The hyper-parameters selected are highly dependent on the model specification and model complexity. These include the distance and accepter functions, the epsilon value at which to terminate, the maximum number of allowed populations and the population size in each SMC ABC step.

1. **The Distance Function** measures the distance of simulated samples to the observed sample. We use the adaptive p-norm distance described by Prangle et al. (2017) which adapts the weights for each of the summary statistics for each generation, based on the relative importance and variation of each respective summary statistic in the previous simulation results. This results in a distance function which is not predefined. Instead, it evolves each iteration depending on the variation of the summary statistics. This is particularly suitable when the summary statistics vary on different scales and it is not clear from the outset how they should be weighted. We use the default, setting $p = 2$ to use the Euclidean (2-norm) distance:

$$d(x, y) = \left[\sum_i \mid w_i(x_i - y_i) \mid^p \right]^{1/p}. \tag{12}$$

   This has two advantages over a non-adaptive Euclidean distance, the first being narrowing densities around the true parameters and the second is faster convergence as the required number of samples is lower due to continually higher acceptance rates (Klinger et al., 2018).

   However, adaptive distance requires an additional step to achieve robustness. Adaptive distances can result in large weights being applied to non-informative summary statistics which have relatively lower variation (Klinger et al., 2018). The solution to this problem is to not only take the in-sample variance but additionally factor in the bias of the samples to the observed data. To compute this we use the approach recommended by Klinger et al. (2018), whereby we customise the distance function to the square root of the mean square error. This equivalent to the bias-squared plus the variance and improves the robustness of the calibration approach.

2. **Accepter Function**: The accepter function, $A(d(s(y_{obs}), s(y)))$, is implemented to determine whether samples from given parameters should be accepted. In the simplest case the acceptance function accepts the sample if the distance between the observed and sample summary statistics is below some ϵ threshold. We make use of the most common accepter function in ABC: the Uniform Accepter. The Uniform Accepter accepts the sample if the distance falls between a Uniform error distribution between $-\epsilon$ and $+\epsilon$ (Wilkinson, 2013).

3. **Acceptance Threshold (ϵ)** is the acceptance threshold which can be predetermined as a fixed number or as a predefined list. Alternatively, ϵ can be determined empirically by a defined strategy for setting a new ϵ for each new generation. We implement the median ϵ strategy whereby each subsequent ϵ is equal to the median distance from the previous population. This is the default epsilon strategy given by pyABC (Klinger et al., 2018).

4. **Terminating Conditions:** There are three terminating conditions for the SMC ABC algorithm as given in Step 3 of Algorithm 2, the minimum epsilon, the minimum particle acceptance rate and the maximum number of populations - whichever is reached first terminates the SMC ABC run. These parameters are highly model-complexity dependent and need to be determined by trial and error. The larger the minimum epsilon, minimum particle acceptance rate and the maximum number of populations the higher the accuracy but the longer the computational time. With a maximum time allowed on UCT's HPC of seven days and maximum number of cores of 40, the highest configurations suitable for the complexity of the PGPS Model (Preis et al., 2006) is minimum epsilon of 0.0001 and maximum number of populations as 6. The populations are generated successively and have a decreased ϵ value with each subsequent population but the acceptance rate also decreases with each

Table 4: Table summarising the algorithmic complexity of each of the perturbation kernels examined by Filippi et al. (2013) given B, the number of particles in the previous population, and dimension p, the number of parameters. For the multivariate Normal based on the Fisher Information Matrix (FIM), C refers to the computational cost of simulating an observation.

Perturbation Kernel	**Algorithmic Complexity**
Component-wise Normal	$O(pB^2)$
Multivariate Normal (based on entire previous population)	$O(p^2B^2)$
Multivariate Normal (based on M nearest neighbours)	$O((p+M)B^2 + p^2M^2B)$
Multivariate Normal with OLCM	$O(p^2B^2)$
Multivariate Normal base on FIM	$O(pCB + p^2B^2)$

population. This is shown in Figure 4 in the first plot as each subsequent colour from bottom to top represents the number of samples required to populate the population in each SMC ABC step ($t = 0, 1, 2, 3, 4, 5$). The Epsilon values plotted alongside it in Figure 4 shows the successive reduction and plateau of ϵ (the acceptance threshold) with each SMC ABC population step ($t = 0, 1, 2, 3, 4, 5$). The minimum particle acceptance rate is set to the population size divided by 25000 so that no more than 25000 simulations of the PGPS Model (Preis et al., 2006) are generated in any given time-step as more than this was computationally infeasible given the above constraints. This is illustrated in Figure 4 which shows that the epsilon values converge in under six populations and the sixth population, which requires approximately 8000 samples, is vastly more computationally expensive than the former populations.

5. **Population Size:** An important parameter is the population size. The accuracy of the calibration is improved by increasing the sample size. However, increasing the sample size also increases the computations required which, inherently, reduces the efficiency of the ABC SMC technique (Klinger et al., 2018). Given the computational complexity of the PGPS Model (Preis et al., 2006) and the restrictions on computing power, a population size of 50 is optimal through determination by trial and error.

6. **Perturbation Kernel:** The final configurable element of SMC ABC is the perturbation kernel which enables the current population to perturb to the subsequent one. Importantly, perturbation kernels are one of the obvious mechanisms of speeding up SMC ABC inference. Filippi et al. (2013) provide and extensive investigation into the appropriate choice of perturbation kernel for SMC ABC. They assess the algorithmic complexity of each of the kernels in Table 4 from a previous population of B particles with dimension p (the number of parameters) and find that a multivariate Normal kernel with OCLM (Optimal Local Covariance Matrix) are preferred for their generally highest acceptance rates and ease of implementation at an acceptable computational cost (Filippi et al., 2013).

3.7 Validation with an ARMA Model

To prove that the SMC ABC calibration technique is applied correctly, we apply this calibration technique to a simple Autoregressive Moving Average (ARMA) model with two parameters. The first parameter, ρ, defines the autoregressive component which explains the momentum and mean reversion effects and the second parameter, ν, defines the shock effects in the white noise term. ARMA models are frequently used to model financial time series (Tsay, 2005). Although they fail to generate volatility clustering, a key stylised fact of financial markets. An ARMA(m,n) refers to the model with m autoregressive terms and n moving-average terms which is formulated as follows:

$$X_t = \sum_{i=1}^{m} \rho_i X_{t-i} + \sum_{j=1}^{n} \nu_j \epsilon_{t-j} + \mu + \epsilon_t. \tag{13}$$

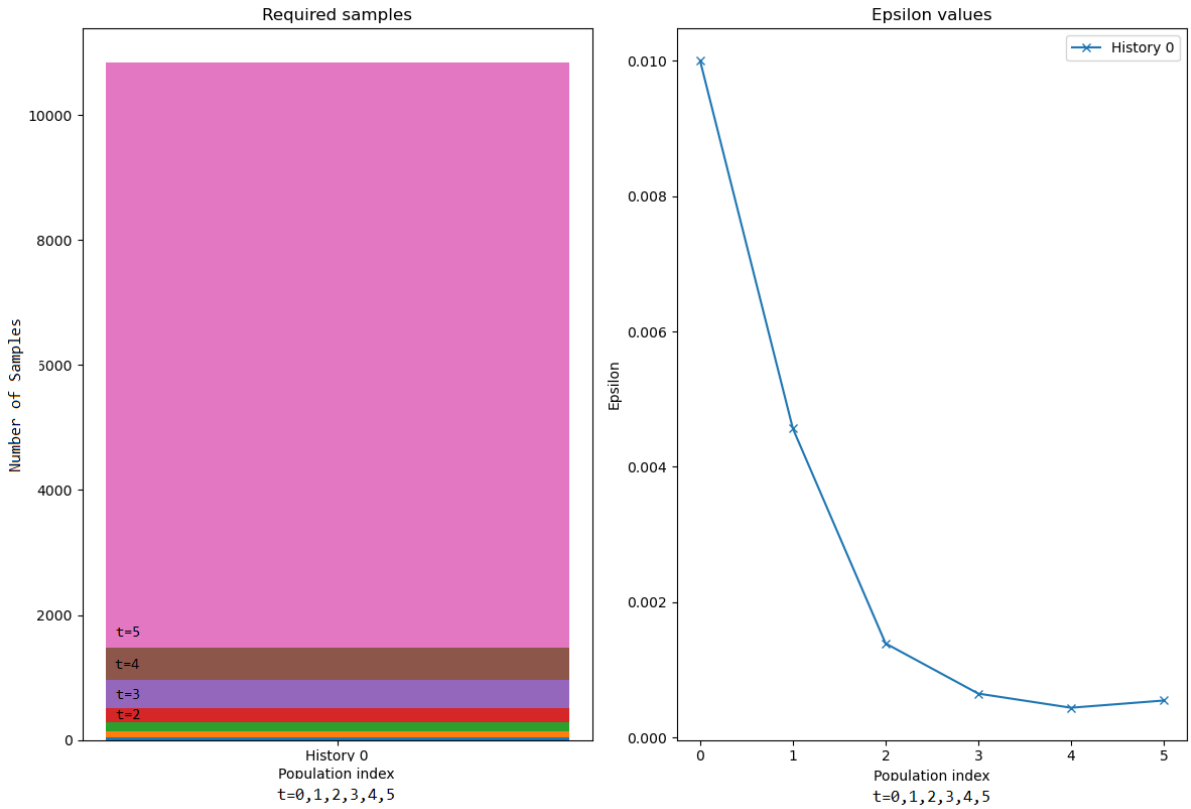

Figure 4: In the first plot each subsequent colour from bottom to top represents the number of samples required to populate the population in each SMC ABC step ($t = 0, 1, 2, 3, 4, 5$). The Epsilon values plot alongside it shows the successive reduction and plateau of the acceptance threshold with each SMC ABC population step ($t = 0, 1, 2, 3, 4, 5$). Both plots showing the terminating conditions determined by trial-and-error. The plot on the left illustrates the number of samples required to generate a population of size 50 in each iteration of the SMC ABC algorithm. The second plot illustrates ϵ decreasing with each subsequent population. This empirical evidence suggests that 6 populations is sufficient and any more would be too computationally costly as the ϵ values plateau around 0.0005 and the number of samples required for the fifth population is vastly larger than what was required for prior populations.

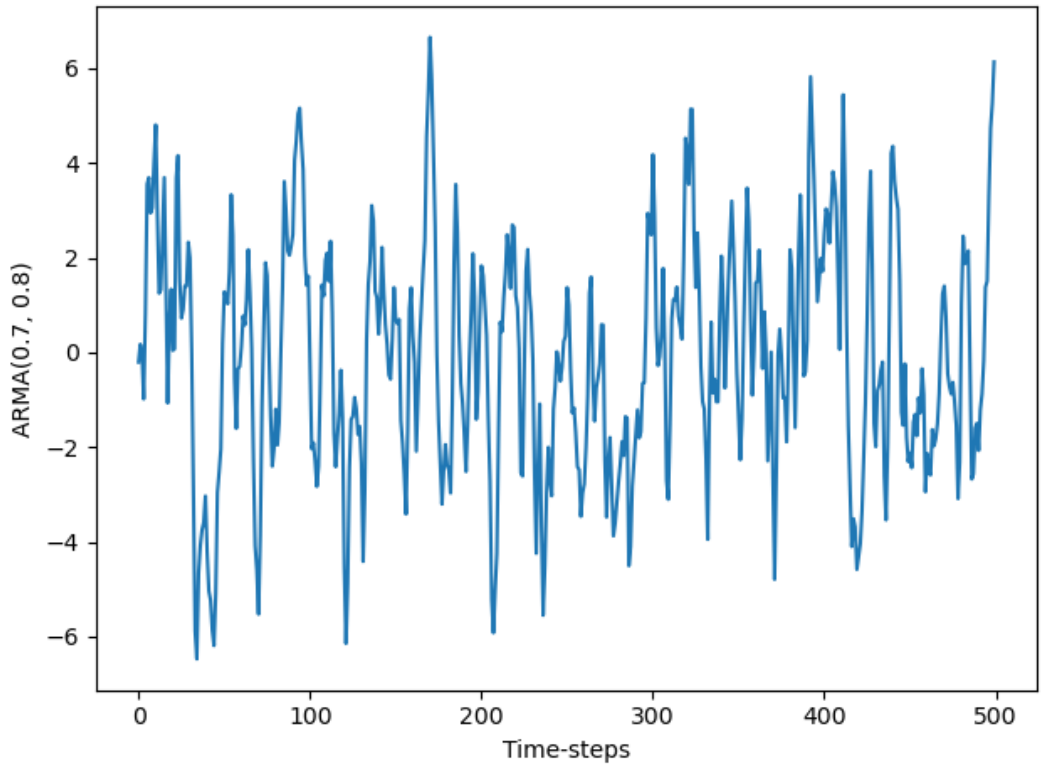

Figure 5: Time-series plot of the simulated "true" ARMA(1,1) time-series of length 500 given true parameters $\rho = 0.7$ and $\nu = 0.8$ for the autoregressive and moving average components, respectively.

Where $\epsilon_t \sim N(0, 1)$ is the white noise term which is independently and identically distributed (i.i.d.) for all t; and $\frac{\mu}{1-\sum_i \rho_i}$ is the expected value of X_t (often assumed to be 0). We are calibrating the simpler AR(1,1) case with mean of 0 which simplifies to:

$$X_t = \rho X_{t-1} + \nu \epsilon_{t-1} + \epsilon_t, \ \epsilon_t \overset{\text{i.i.d.}}{\sim} N(0, 1) \ \forall t \in 0, 1, 2, ... \tag{14}$$

To calibrate this model using SMC ABC we simulate an observed "true" time series from the ARMA(1,1) model with true parameters $\rho = 0.7$ and $\nu = 0.8$, illustrated in Figure 5. For simplicity of this toy-problem, we assume a non-informative uniform prior where both $\rho \sim U(0, 1)$ and $\mu \sim U(0, 1)$. The summary statistics initially used are given in Table 3. However, the calibration performed poorly on this set of summary statistics. To attempt to improve the calibration, the auto-correlation with 1-5 lags was added to the original set of summary statistics. The hyper-parameters used are identical to those described in Section 3.6.5 that are used for the PGPS Model (Preis et al., 2006) calibration.

The results of the ARMA calibration were successful in that, with increasing number of populations, the estimates of ρ and ν converged to their true values and the level of confidence increased. This is visualised in Figure 6. This simple SMC ABC example provides us with two key points, firstly, the successful calibration proves that the implementation of SMC ABC has been done correctly and secondly, the success of the calibration is highly dependent on the summary statistics selected. Before incorporating the lagged auto-correlations the calibration performed poorly but, adding these in, significantly improved the calibration. Another point worth noting is that even though a finite set lags of the sample ACF is indeed not a sufficient statistic for ARMA parameter estimation, we were still able to successfully calibrate the ARMA(1,1) model using SMC ABC (Bruzzone and Kaveh, 1984).

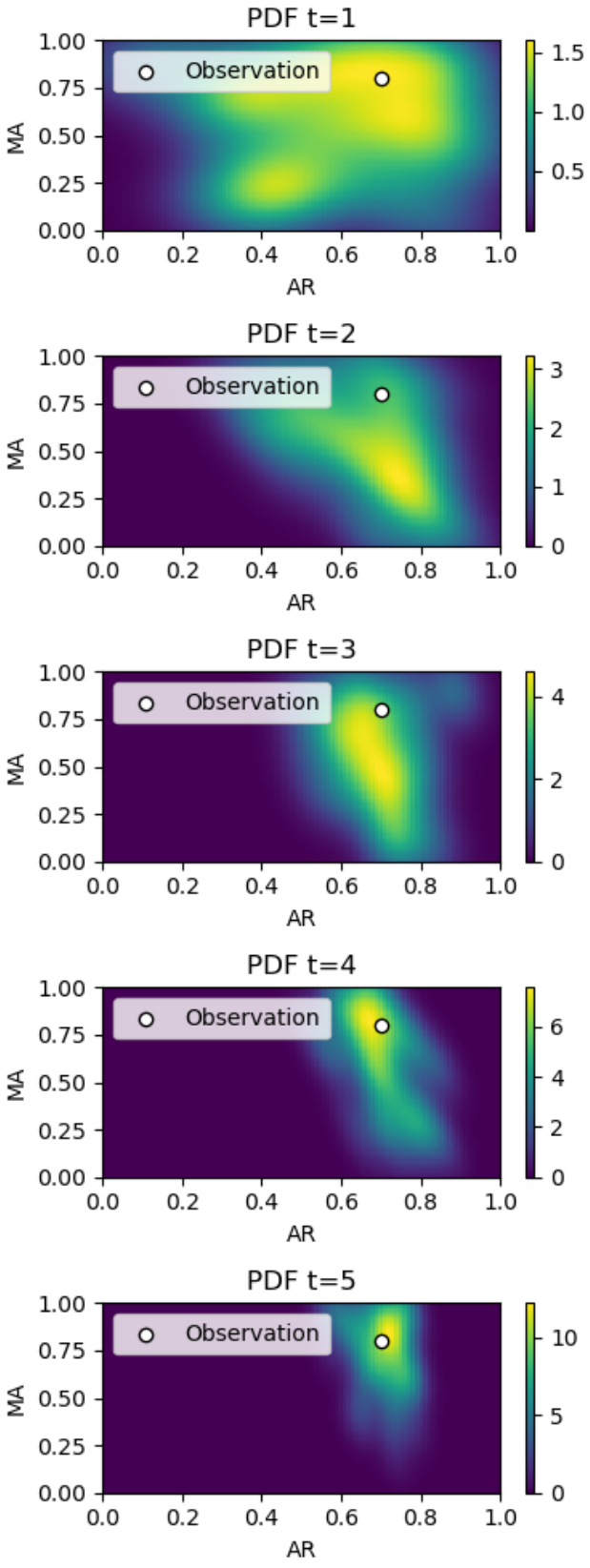

Figure 6: Figure showing successive improvements in calibration with each population posterior Probability Density Function (PDF) of the SMC ABC process to fit the ARMA(1,1) model. The plots are enumerated from $t = 0, 1, 2, 3, 4, 5$ for each succesive population step in the SMC ABC algorithm as each subsequent plot represents a movement towards to the true underlying parameters. Each plot represents the numeric joint probability density function for the AR and MA parameters with the white circle observation representing the true value of the parameters. This illustrates 5 iterations (populations) required for the SMC ABC algorithm to converge to the true parameters.

3.8 Likelihood Inference

The primary concerns with SMC ABC calibration are its inherent need for summarising the model output into what are generally insufficient summary statistics and secondly the rejection step. In contrast, likelihood theory, which has had tremendous success in parametric inference has the benefit of directly optimising the likelihood of the output data given the parameters (Fan et al., 2001).

In terms of aggregating the data into summary statistics, for models with exponential family distribution, likelihood theory enables the use of sufficient statistics, such as the sample mean and sample variance for a Normal distribution, to directly estimate the true underlying mean and variance (Beaumont et al., 2009). This implies that the sufficient statistics can estimate the parameters equally as well as if all the data points were provided. Trivially, when performing inference on models where the sufficient statistics are known, they should be used in both likelihood and ABC paradigms. However, for intractable models where ABC is generally used over likelihood this is unlikely to be the case as Pitman-Koopman-Darmois theorem informs that summary statistics are only bounded irrespective of the sample size in the simple case where the model is in the exponential family (Beaumont et al., 2009). Consequently, ABC generally requires a set of summary statistics that are not necessarily sufficient. In the autoregressive moving average model example provided in Section 3.7, there are no sufficient statistics that can be computed to summarise all the information about the parameters in the data. As such, the success of the calibration is shown to be heavily dependent on the summary statistics used because the first configuration fails to calibrate the parameters, whereas, the second set of summary statistics which provided additional information in the summary statistics selected, resulted in a successful calibration.

Additionally, the acceptance-rejection mechanism used in ABC is another area where information is lost in comparison to the likelihood optimisation approach (Platt, 2020). For candidate parameters where the parameter is rejected, we lose all information about that parameter rather than incorporating it into a robust distribution with a low probability as given in the likelihood distribution. This adds to the computational complexity of the ABC algorithm as we require more computations to gather up information that has been lost and throws away information about unlikely but possible parameter estimates.

Recently, Platt (2020) has provided an in depth exploration and comparison of a number of calibration techniques in both Frequentist and Bayesian paradigms. In this research, Platt (2020) identifies the outperformance of Bayesian approaches, such as ABC, over Frequentist approaches across a range of models from simple, to highly complex. However, for highly complex models, it is the likelihood nature of the calibration technique which drives the success of the calibration independently of whether it was implemented in a Frequentist or Bayesian approach. In particular, he found that for more complex models with large parameter spaces which inherently require more information to separate out the effect of each parameter, the success of the calibration was highly dependent on the incorporation of likelihood inference. The calibration technique which proved most successful in this research across models of various complexity is the Bayesian likelihood approach introduced by Grazzini et al. (2017) (Platt, 2020). This further illustrates the limitations of the SMC ABC algorithm and the information loss associated with the summary statistics used in the acceptance-rejection scheme and leads towards future research in Bayesian likelihood techniques which are out of scope of this research.

4 Calibration Results

4.1 Synthetic Calibration

Noting the importance of summary statistics in the ARMA calibration, we compare the results of two configurations of summary statistics for the SMC ABC calibration. The first configuration uses the same

default summary statistics as given by Platt and Gebbie (2018), the mean, standard deviation, skewness, kurtosis, Hurst exponent and Kolmogorov Smirnov statistic (detailed in Table 3) and the second, uses these same statistics along with five lags of auto-correlation. We then plot the joint and individual posterior distributions of the parameters as well as their progression towards those distributions with each subsequent SMC ABC iteration. We compare the posterior credibility intervals to the known underlying parameters that were used to generate the synthetic data to assess the success of the calibration.

4.1.1 Summary Statistic Configuration 1

Using the SMC ABC algorithm with hyper-parameters described in Section 3.6 and the five summary statistics given in Table 3, we plot the resulting joint and univariate distributions of all six parameters in Figure 7. From these distributions, it is evident that convergence has not been attained for parameters α and δ due to their flat posterior distributions and, as such, we can have little to no confidence in their posterior estimates. As for the remaining parameters, only λ_0 has come close to converging to its known underlying value. The remaining parameters have converged to values that do not align with the known underlying synthetic parameters given in Table 1.

Additionally, we observe the instability of posterior distributions and their lack of convergence in Figure 8 under the trajectory of iterations through the SMC ABC calibration. The distributions vary widely from each subsequent population step to the next with no indication of convergence to a true underlying parameter. This is in stark contrast to the convergence that can be seen in each subsequent iteration of the calibration given in Figure 6 for the autoregressive moving average model and the successful calibration for its two parameters.

There are two broad reasons why the parameters fail to converge to the true underlying values, the first is due to parameter identifiability and second is due to information loss in the SMC ABC calibration.

1. **Parameter Identifiability:** An identifiable parameter is one in which the parameter provides information that is evident in the data output. Parameters may not be identifiable for three reasons. Firstly, if the parameter does not affect the model output at all, then its value cannot be uniquely determined as any value would lead to the same model output. Secondly, if the parameter in fact does have meaningful impact on the model output but is collinear with another parameter then neither parameters can be uniquely determined unless one of them is fixed - this results in an infinite number of possible combinations of these parameters that would result in the same output. Thirdly, if the parameter's effect on the model output is weak relative to the noise of the output, it will be impossible to determine a unique value. Parameter identifiability is a known issue particularly for models with many parameters and relative complexity and the PGPS Model (Preis et al., 2006) is one such case Platt (2020). This may be a reason why this particular model has never been successfully calibrated and why similarly complex agent based models for market micro-structure data (which is known for its noisy data) have never been successfully empirically calibrated (Platt and Gebbie, 2018). As described in Section 3, this is likely why stylised fact-centric model validation has dominated agent based model literature for financial markets particularly for more complex ABMs with large parameter spaces as the more parameters there are in the model, the more combinations of parameters can be entered to suitably "fit" the stylised facts (Panayi et al., 2012) (Guerini and Moneta, 2017). This results in overly complex models where the parameters are not uniquely identifiable. This explains why SMC ABC is successful in calibrating a simple autoregressive moving average model as shown in Section 3.7 but fails to calibrate a more complex model with more parameters and noisier data, the PGPS Model (Preis et al., 2006).

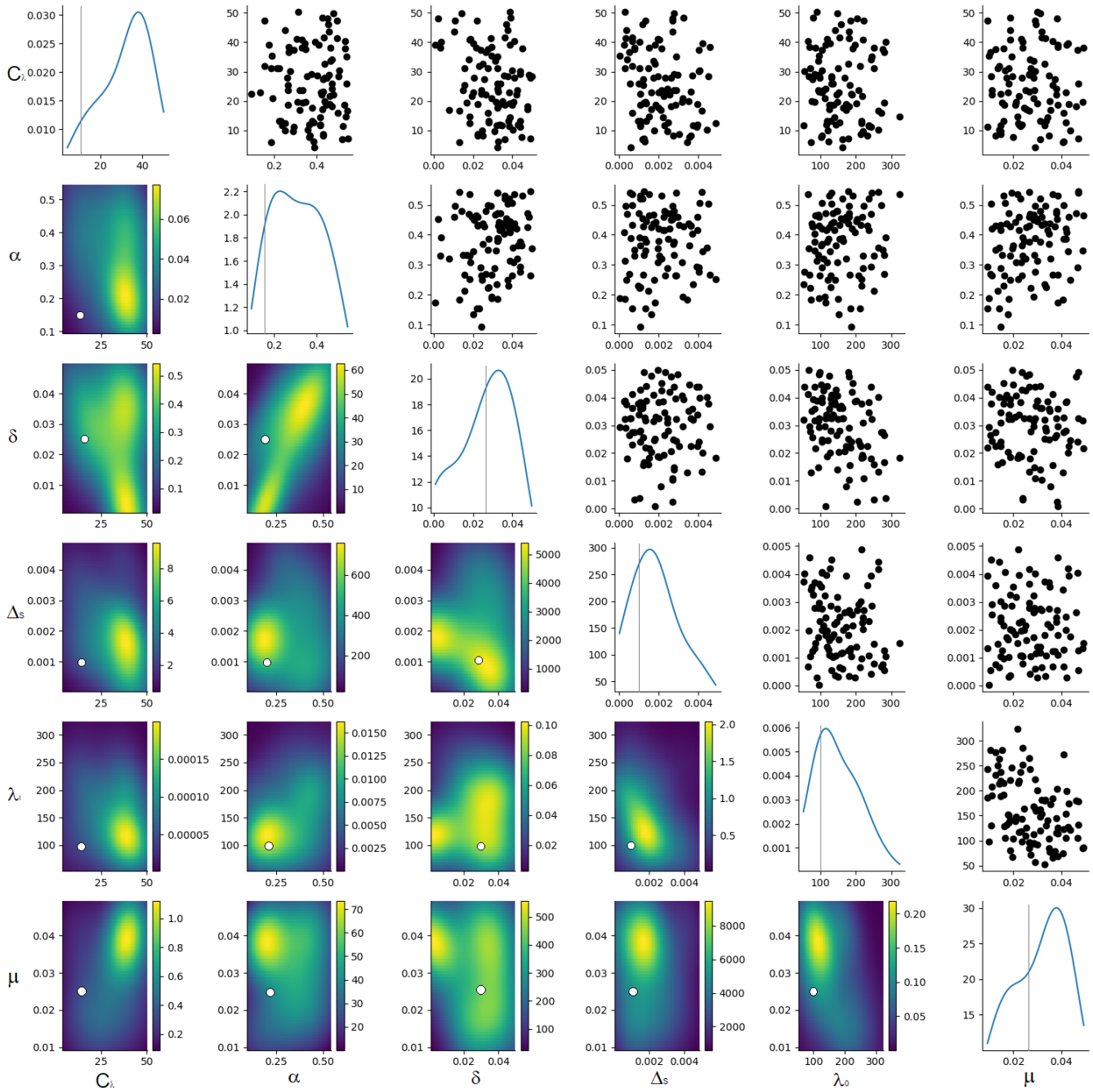

Figure 7: Synthetic calibration with summary statistic Config-1: Joint and individual posterior density functions of each of the six parameters for SMC ABC with the first parameter configuration (6 summary statistics used under Platt and Gebbie (2018). See Section 4.1.1 for information on summary statistics used. Calibration is poor as only a single parameter, λ_0, converges close to its true value. The circles and grey lines represent the true underlying parameters.

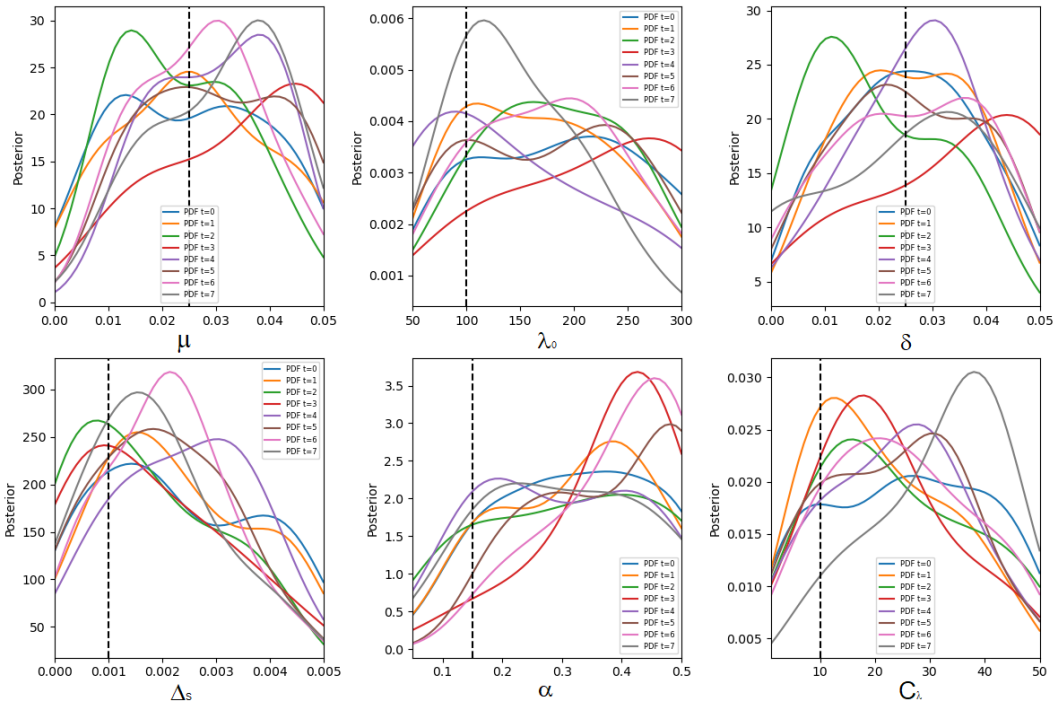

Figure 8: Synthetic calibration with summary statistic Config-1: Figure showing univariate posterior density functions for the six parameters in six plots and each of their successive SMC ABC steps enumerated $t = 0, 1, 2, ..., 7$ (see Figure 6 for example of successful iterative movement towards the true parameter estimates with each SMC ABC step). See Section 4.1.1 for information on summary statistics used. There is no evidence of a successive progression towards the true parameter set as the last two population density iterations ($t = 6$ and $t = 7$, given in pink and grey, respectively) fail to have the highest density around the true parameter values. This is with the exception of λ_0 which on its $t = 7^{th}$-step has the highest density around its true value. However, there is no evidence of it successively converging since λ_0 in its previous step does not have high density around this value and, therefore, the appeared convergence to the true value of λ_0 may be due to randomness. The true underlying parameters are given by the vertical dashed lines.

2. **SMC ABC Information Loss:** Assuming the models parameters are all uniquely and jointly identifiable given the model output then the remaining reason for why the parameters cannot be calibrated using SMC ABC is because there is information loss in the process of using summary statistics under SMC ABC. Subsequently, in any ABC application, the results are highly dependent on the summary statistics selected (Prangle et al., 2017). This was particulary evident in our toy ARMA model which failed to calibrate given the first parameter set but calibrated well when using an enhanced set of summary statistics. The higher the information loss from the model output to the summary statistics, the less likely the success of the calibration. However, choosing summary statistics is a necessity for calibrating computationally expensive models, such as the PGPS Model (Preis et al., 2006), using computationally expensive calibration techniques, such as ABC. This is because aggregating the model output vastly improves the computational efficiency of the ABC algorithm by improving the efficiency of the comparison between true observations and simulated data. The choice of summary statistics is particularly challenging since the best choice of summary statistics varies across data sets (as noted by Joyce and Marjoram (2008) in accordance with Pitman-Koopman-Darmois theorem (Nunes and Balding, 2010). Subsequently, the SMC ABC calibration of the PGPS Model (Preis et al., 2006) could be failing for the same reason as the simulated method of moments calibration method failed for Platt and Gebbie (2018): there is material information loss in summarising the data into the six summary statistics provided in Table 3. These summary statistics may have reduced the information provided in the model data enough to remove information about one or more of the underlying parameters. This is a key difference between maximizing a likelihood versus using a method like SMC ABC which requires summary statistics of the data. Even if the parameters had weak effect on the model output, summarising the output may reduce the information such that the parameter is no longer identifiable. As a result we trial the second configuration of summary statistics, adding five lags of auto-correlations to the initial summary statistics set used by Platt and Gebbie (2018).

4.1.2 Summary Statistic Configuration 2

The second configuration of summary statistics incorporates all the summary statistics given in Table 3 as well as an additional five statistics for the first five lags of the auto-correlations. This calibration converges for all six of the underlying parameters but converges to incorrect values for four out of six parameters. It successfully calibrates the two parameters, $\delta = 0.025$ and $\mu = 0.025$, to their true underling values but incorrectly estimates the remaining four parameters. Adding the auto-correlations seems to slightly improve the calibration performance of the second configuration of summary statistics over the first but certainly, parameter identifiability or information loss is still resulting in unidentifiable parameters and an unsuccessful calibration. Additionally, from the plots of the univariate distributions for each successive population shown in Figure 10 we can see that there is no evidence of progression towards successfully calibrating the true parameters, with the exception of parameter δ and μ.

We proceed to calibrating real-world high frequency trading data in the Section 4.2 with cynicism. Based on the calibration results on the synthetic output we are not confident that SMC ABC will be able to converge to a set of underlying parameters. Even if convergence is achieved for the real-world high frequency trading data, we do not anticipate that the posterior estimates will be uniquely identifiable. We will examine whether there is a consistent drive to the true parameter as SMC ABC iterates through each population to identify whether the parameter has converged randomly or uniquely. We proceed using only the second configuration of summary statistics as it provides more information and had slightly superior calibration results over the first configuration of summary statistics.

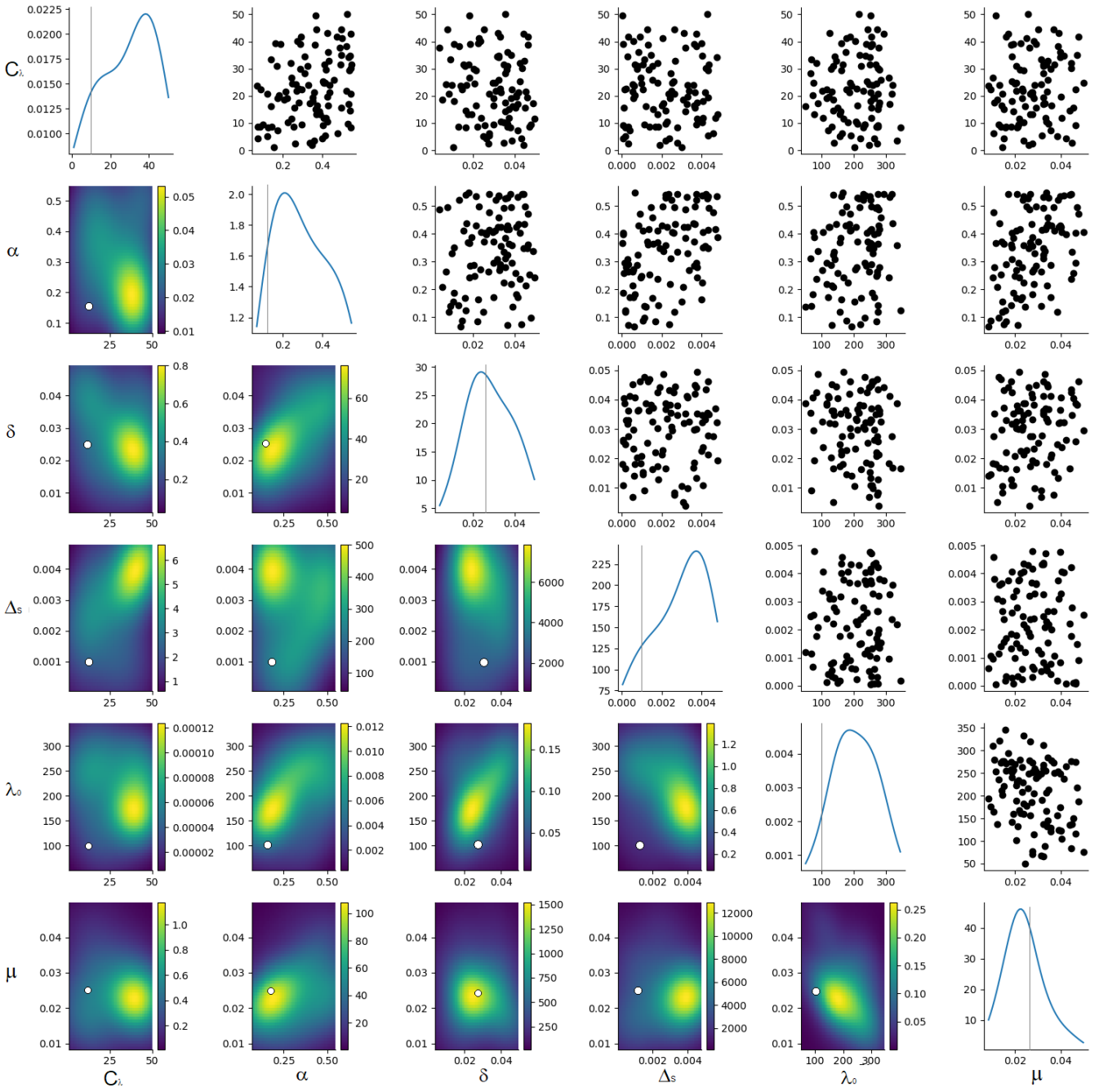

Figure 9: Synthetic calibration with summary statistic Config-2: Joint and individual posterior density functions of each of the six parameters for SMC ABC with the second parameter configuration (6 summary statistics used under Platt and Gebbie (2018) and additionally the first five auto-correlations). See Section 4.1.2 for information on summary statistics used. Calibration remains poor with only two parameters, δ and μ, appearing to converge to their true values. The circles and grey lines represent the true underlying parameters.

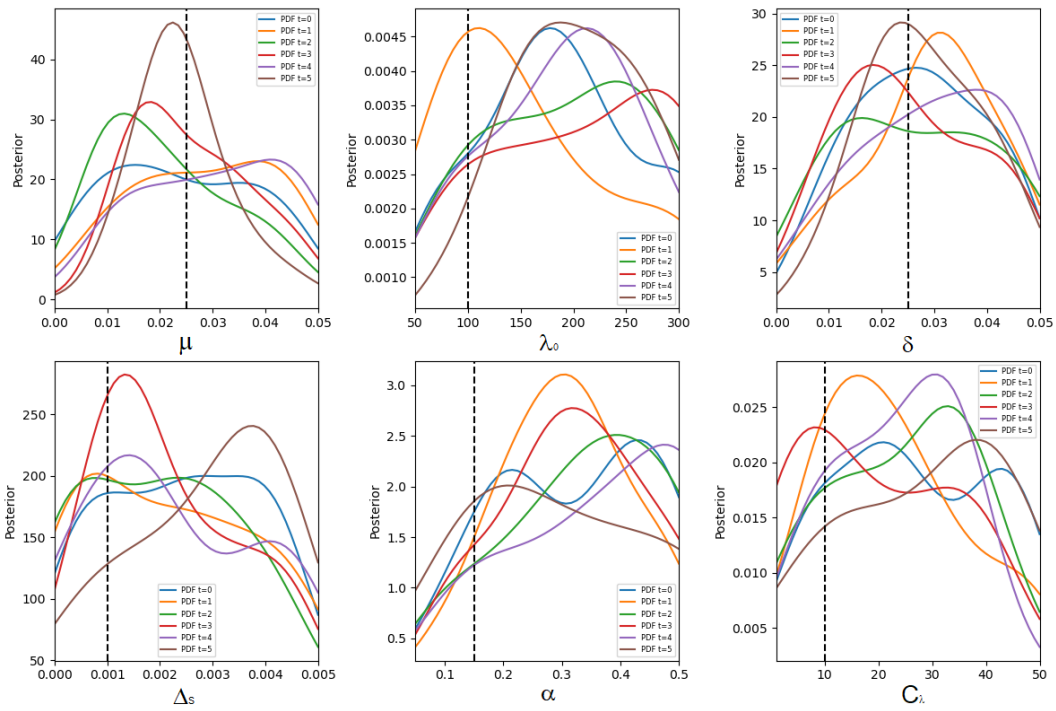

Figure 10: Synthetic calibration with summary statistic Config-2: Figure showing univariate posterior density functions for the six parameters in six plots and each of their successive SMC ABC steps enumerated $t = 0, 1, 2, ..., 7$ (see Figure 6 for example of successful iterative movement towards the true parameter estimates with each SMC ABC step). See Section 4.1.2 for information on summary statistics used. There is no evidence of a successive progression towards the true parameter set with the exception of δ and μ which indeed have highest density around their true values in the final SMC ABC iteration ($t = 5$). However, it appears that the successful convergence of μ may be due to randomness as previous densities, notably $t = 4$, do not have high densities around the true value, whereas, δ appears to successfully transition to higher density around the true value. The true underlying parameters are given by the vertical dashed lines.

4.2 Calibration on JSE Time Series Data

In this section we provide detail on the market micro-structure JSE data used, the posterior estimates of the parameters calibrated using the real-world high frequency trading data and the extent to which the parameters have converged to underlying values. Finally, we assess whether the simulated price path using the calibrated PGPS Model (Preis et al., 2006) is able to recover important stylised facts and we compare this to the stylised facts of the real-world data to which the simulated price path was calibrated.

4.2.1 Data Requirements Specification

The real-world data being used as the observations in the calibration is a week of trading an individual liquid stock which is listed on the JSE, Anglo American PLC (Gebbie and Platt, 2019). We use the week period from 9:10 on 1 November 2013 to 16:50 on 5 November 2013. Our consideration of this dataset is motivated by the fact that calibration of PGPS Model (Preis et al., 2006) was performed over this time horizon for this particular stock by Platt and Gebbie (2018) using simulated method of moments and we want to compare the SMC ABC calibration to their calibration on an equal footing.

The intra-day data obtained contains continuous trading from 09.00 till 16.50, with opening and closing auctions from 08.30-09.00 and 16.50-17.00, respectively. The opening and closing auction data needs to be removed and, to reduce the noise resulting from the opening auction, the first ten minutes of continuous trading are also filtered out of the data. The markets are extremely illiquid in the sequence of trades just after market opening and there is gaming or timing risk when the market switches from continuous trading to closing auction. Therefore, to remove this noise, we subset our intra-day trades to those trades between 09h10m00 and 16h49m59s.

The data used in this calibration research is acquired in Thomas Reuters Tick History (Thomsonreuters.com, 2019) format, whereby the data is formatted into tick-by-tick time-series of quotes and trades. The actual sampling frequency is given in seconds which we convert to a series of 60-second price bars where each price corresponds to the final quoted mid-price at the end of each minute. The mid-price quote is the average level 1 ask and bid price associated for that quote. From this series, we obtain a series of log prices which is the observable series to which we will calibrate the model. A unit time interval is defined as a minute bar. One-week periods are used to calibrate the ABM, which corresponds to 2300 one-minute price bars per week and 460 one-minute price bars per day.

4.2.2 Empirical Calibration

Using the JSE data described in Section 4.2.1, we calibrate the PGPS Model (Preis et al., 2006) using the summary statistics described in Section 4.1 Configuration 2. We anticipate from the synthetic calibration results discussed in Section 4.1 that even if the parameter estimates calibrated have a fairly narrow credibility interval, we cannot assume that these parameters are uniquely identifiable and, therefore, the true underlying parameters cannot be used to infer causal effects from the parameters on these models (LeBaron, 2006), (Barde, 2016) (Lamperti, 2015). To reinforce this, we examine Figure 11 which shows the successively calibrated univariate distributions for all of the parameters for each iterative step in the SMC ABC calibration. The distributions vary widely with no indication of successive convergence and increased confidence towards an underlying "true" parameter for all parameters with the exception of δ. This indicates that the parameters are not uniquely identifiable given the information provided by the summary statistics as the posterior distributions are inconsistent with subsequent populations. This is as expected given our synthetic calibration results in Section 4.1. As such, we select the posterior distribution with the smallest distance from the observed JSE data summary statistics and provide the posterior estimates along with their standard deviations and credibility intervals in Table 5 but we assume that these are not

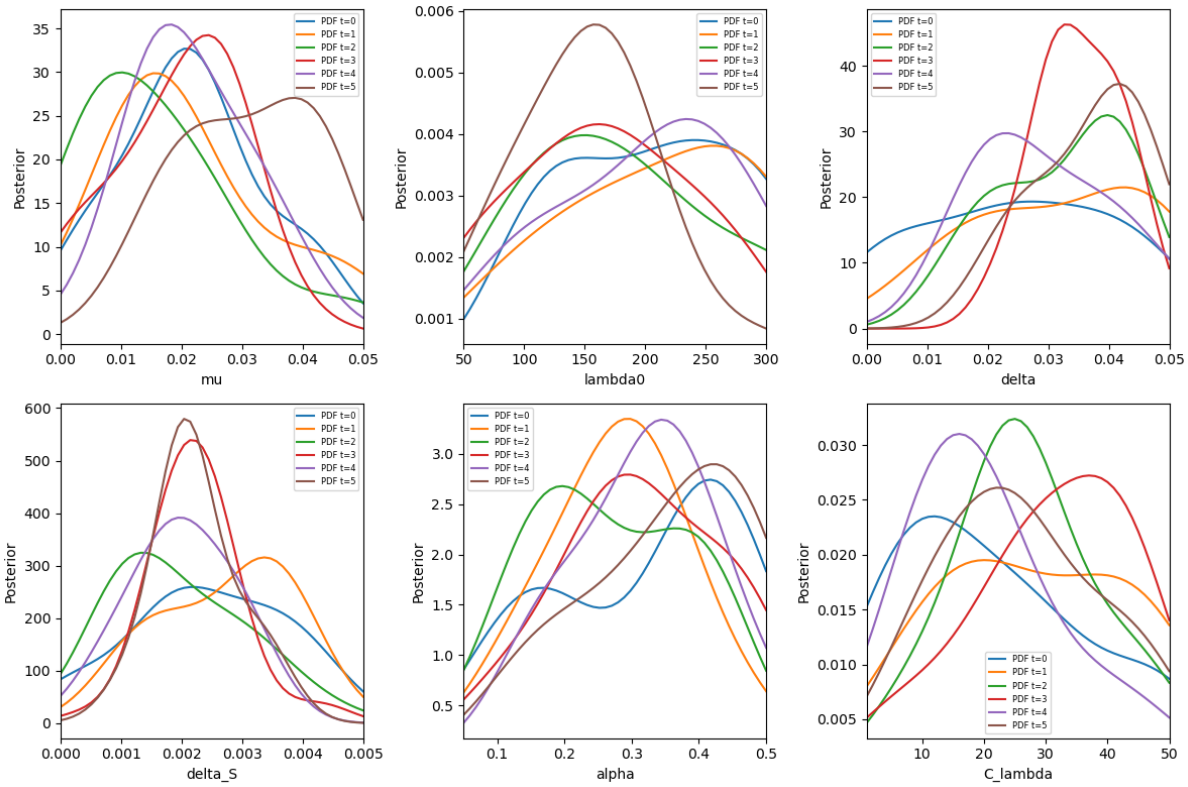

Figure 11: Real-world calibration with summary statistic Config-2: Figure showing univariate posterior density functions for the six parameters in six plots and each of their successive SMC ABC steps enumerated $t = 0, 1, 2, ..., 7$ (see Figure 6 for example of successful iterative movement towards the true parameter estimates with each SMC ABC step). See Section 4.1.2 for information on summary statistics used. The calibration observations used as the underlying data are Anglo American PLC high frequency trading data. There is no evidence of a successive progression towards a true uniquely identifiable parameter set with the exception of Δ_S which appears to successfully transition to higher density around the true value. The remaining parameter posterior distributions vary widely with each successive step of the SMC ABC algorithm with no indication of movement towards a stable distribution with high density around a posterior estimate.

the only possible combination of parameters that could simulate a price path with similar properties to the Anglo American PLC data.

4.2.3 Stylised Fact Validation

In this section we validate the calibrated model described in Section 4.2.2 using stylised fact-centric model validation. We discuss and empirically detect whether stylised empirical facts of intra-day financial markets have been realised for the simulated data as they appear in the Anglo American PLC real-world data. To do this we simulate market micro-structure for a week of trading using the PGPS Model (Preis et al., 2006) and the parameters calibrated using Anglo American PLC trading data (provided in Table 5). In figures 12 and 13 we plot the mid-price path and associated returns and log-returns of the real-world and simulated trading data, respectively. We can then visualise and quantify the returns, distributions and auto-correlations for both the real-world data and simulated data to ascertain whether the calibrated PGPS Model (Preis et al., 2006) successfully reproduces key stylised facts.

By stylised facts, we mean the nontrivial statistical properties that have persisted across a broad range of financial instruments and markets for more than 50 years (Staccioli and Napoletano, 2020). We differentiate stylised facts into two relevant categories: time-invariant stylised facts and intra-day stylised facts. The former applies to financial time series across various time-scales whereas the latter are intra-day or market

Table 5: Posterior mean estimates and 90% credibility intervals for the SMC ABC calibration of the six PGPS Model (Preis et al., 2006) parameters using Anglo American PLC high frequency trading data over the period of a week (9:10, 1 November 2013 - 16:50, 5 November 2013).

Parameter	**Posterior Mean**	**90% Credibility Interval**	$\frac{\mathbf{s}}{\sqrt{\mathbf{n}}}$
μ	0.015	(0.0128,0.0189)	0.0003
λ_0	180.4571	(158.6849,202.2293)	2.4022
δ	0.0361	(0.0337, 0.0385)	0.0003
Δ_S	0.0022	(0.0019,0.0025)	0.0001
α	0.2978	(0.2606,0.3349)	0.0041
C_λ	30.2497	(26.7511, 33.7483)	0.3860

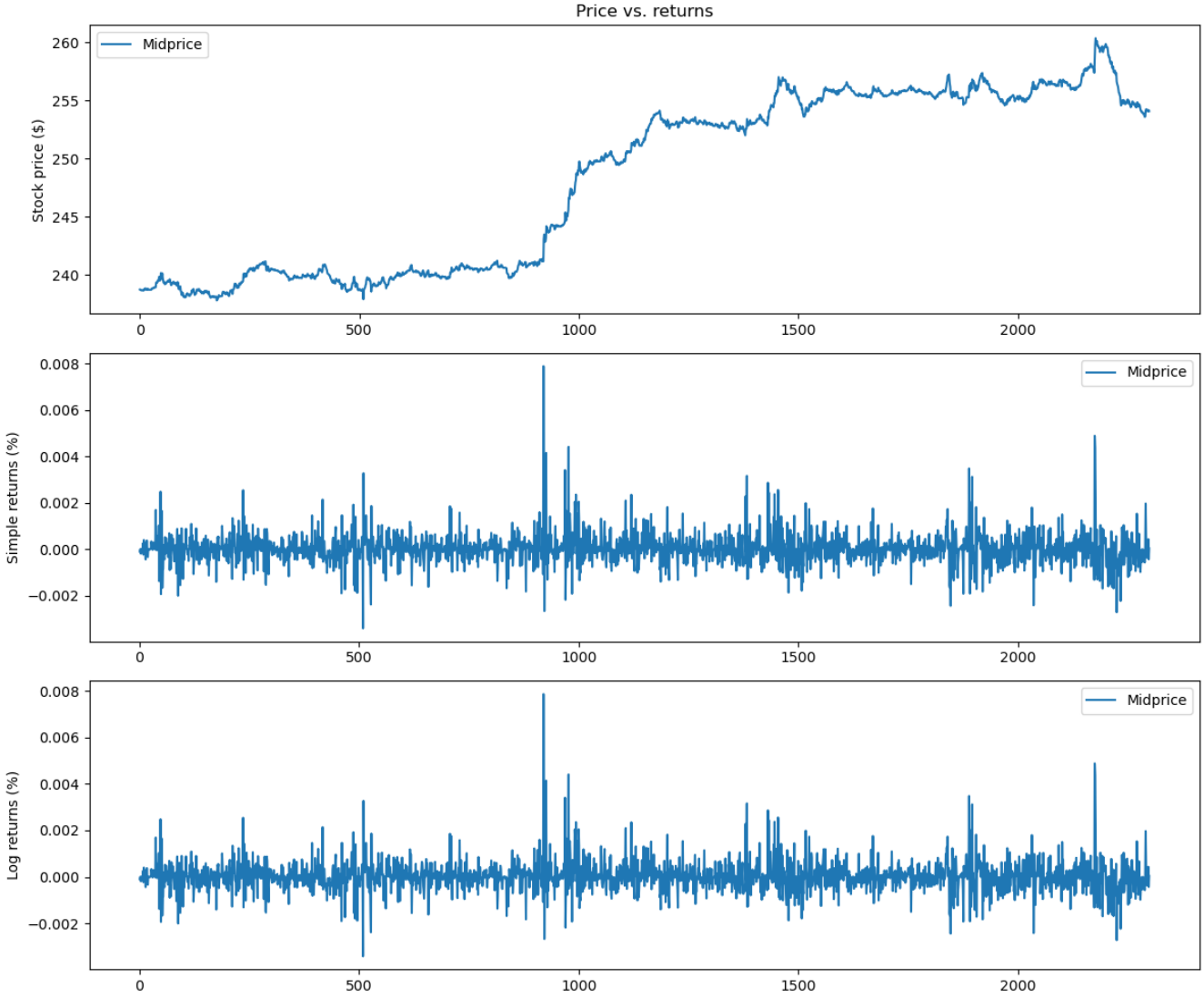

Figure 12: Plots illustrating the mid-price and associated returns and log-returns of real-world Anglo American PLC shares over a week of trading.

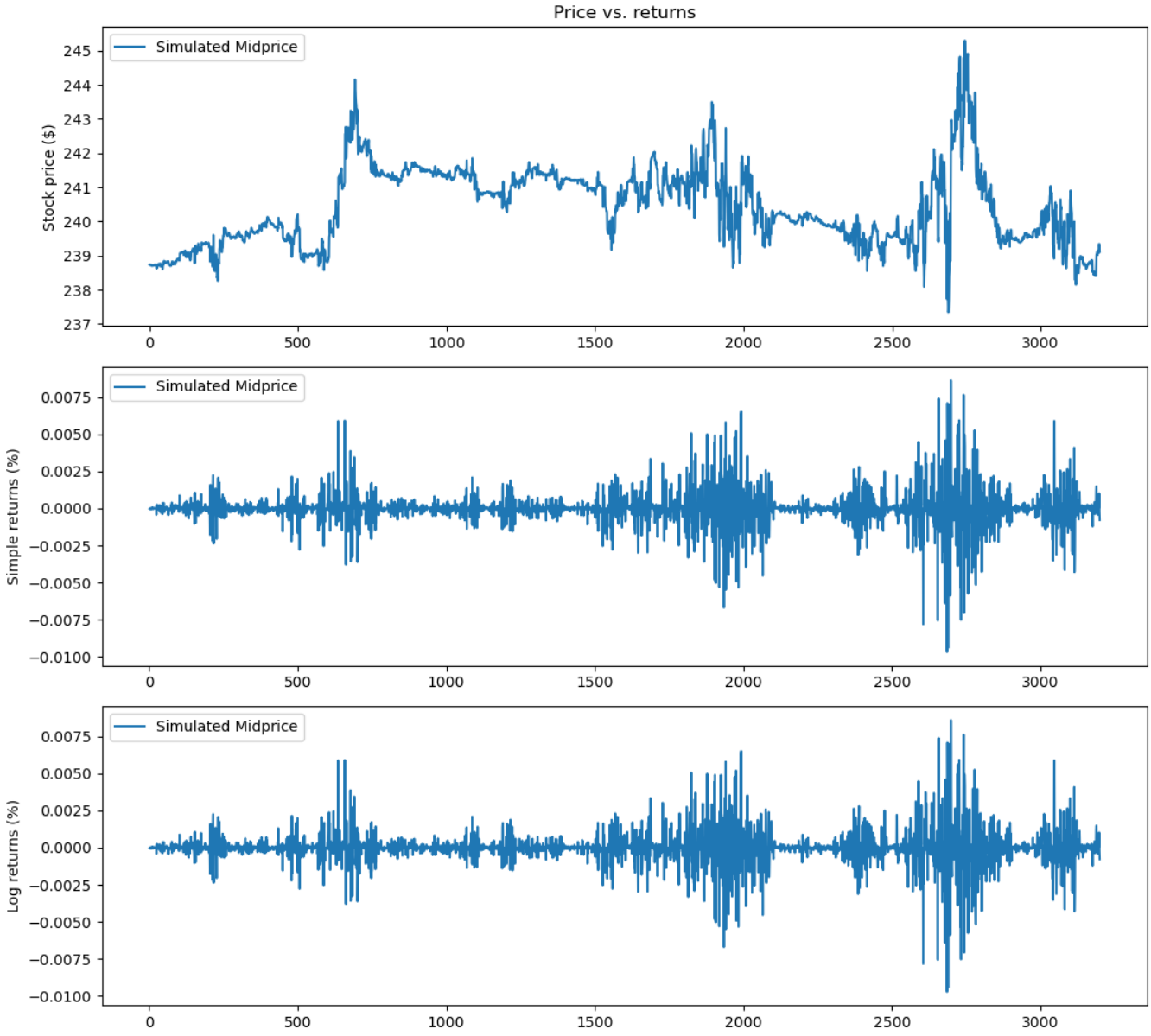

Figure 13: Plots illustrating the simulated mid-price, associated returns and log-returns over a week of trading. Simulations were obtained using the PGPS Model (Preis et al., 2006) and calibrated parameters given in Table 5. See Section 4.1.2 for information on summary statistics used.

micro-structure specific. The time-invariant stylised facts we investigate in this research are leptokurtic returns, absence of auto-correlation of raw returns and volatility clustering. We assess order-flow clustering as a key intra-day stylised fact.

1. **Leptokurtic Returns:** Under various leading top-down models, such as Black-Scholes derivative pricing, returns of financial instruments are specified as log-Normally distributed (Black and Scholes, 1973). However, this is in contrast to reality whereby the distribution of log-returns is leptokurtic in relation to the Gaussian distribution, i.e. log-returns exhibit higher kurtosis than the Normal distribution (Staccioli and Napoletano, 2020). This implies that more extreme returns are more likely than is indicated by the log-Normal distribution which results from more extreme reactions to positive or negative information entering the market. Figure 14 shows the log-return distributions relative to the Normal distribution and QQ-plots comparing the log-return sample quantiles to that of theoretical Normal qunatiles for both the real-world and simulated returns. It is evident from both the distribution and QQ-plots that both the real-world and simulated returns are leptokurtic as there is evidence of heavier tails in the QQ-plot and a higher peak which is evident from the distribution plot. This is a promising feature of the PGPS Model (Preis et al., 2006)'s ability to recover leptokurtic returns. However, the degree of kurtosis appears to be exaggerated for the simulated log-returns in comparison to the real-world log-returns to which it was calibrated indicating, as anticipated, that the calibration using SMC ABC was unsuccessful.

2. **Absence of auto-correlation of log-returns:** Intuitively, if subsequent returns were statistically correlated with previous returns, then one could trade using the previous returns to generate risk free trading profit. Arbitrage pricing theory implies this cannot hold as arbitrageurs would identify and trade on this knowledge until effectively the arbitrage is priced out (Roll and Ross, 1980). This implies that auto-correlations of returns in the real-world quickly decay to zero (Staccioli and Napoletano, 2020). We can observe this by looking at the first auto-correlation plot in Figure 15 where all lags of auto-correlations are not statistically different from 0 at the 95% confidence level for the real-world data. We then refer to the return auto-correlation plot given in Figure 16 for the simulated data and observe significant negative auto-correlations of returns for lags 1, 3 and 5. This implies the calibrated PGPS Model (Preis et al., 2006) fails to recover this important stylised fact of finacial markets: lack of auto-correlation of returns.

3. **Volatility Clustering:** Although the raw returns should not be auto-correlated, a key stylised fact is positive auto-correlations of non-linear functions of the returns (Staccioli and Napoletano, 2020). This implies that although the signs of returns are not predictable, the magnitude of returns is inherently predictable in financial markets and that there is persistence in the magnitude of price changes or returns. In literature, this is described as persistent volatility shocks whereby large price movements are more likely to be followed by similarly large movements and small price movements are followed by similarly small movements in the price. In the return plots in Figure 12 for the real-world returns and Figure 13 for the simulated returns, there is clear evidence of volatility clustering in that volatility clusters are formed in the time series. However, a more robust approach to analysing volatility clustering is to look at the auto-correlations of non-linear functions of the log-returns, i.e. the squared returns, r_i^2, or absolute returns, $\mid r_i \mid$. From the auto-correlations of the squared and absolute returns provided in figures 15 and 16 we can see that that magnitude of returns are positively auto-correlated for both the real-world and simulated returns. Indicating that the calibrated PGPS Model (Preis et al., 2006) is able to recover another important stylised fact although it does seem to exaggerate the effect of volatility clustering in comparison to the real-world returns to which it was calibrated.

4. **Order Flow Clustering:** An important intra-day stylised fact is order-flow clustering which means that the side of order book transactions are positively auto-correlated (Biais et al., 1995). This implies that buy-orders will follow buy-orders more often then sell-orders and that sell orders will follow sell-orders more often than buy-orders. To test whether the simulated order book recovers order-flow clustering we need to plot the auto-correlations of the trade signs for the simulated data. Although, by construction of the PGPS Model (Preis et al., 2006) we inherently have the trade signs, in order to fairly compare this to the real-world data for which the underlying trade signs are unknown, we estimate the trade signs using the tick rule which uses the price changes of previous trades. The tick rule classified trades as hitting the sell (buy) -side of the order book if the transaction is above (below) the previous price. Similarly, the trade is classified as hitting the sell (buy) -side if there is no price change but the previous tick change was up (down) (Biais et al., 1995). We plot the auto-correlations of these trade signs obtained from the tick rule in the fourth plot in figures 15 and 16 for the real-world and simulated trades, respectively. There is evidence of significant positive auto-correlation of trade signs for a lag of 1 for Anglo American PLC indicating order-flow clustering exists. In contrast, the order-flow auto-correlation plot in Figure 16, shows no evidence of positive statistically significant auto-correlation of trade signs indicating that the PGPS Model (Preis et al., 2006) has failed to recover the important intra-day stylised fact of order-flow clustering. This is in contradiction with one of the supposed benefits of the PGPS Model (Preis et al., 2006) which was designed to recover order-flow clustering. However, when the parameter is not chosen arbitrarily to achieve this feat but is rather calibrated using real life data, the model does not recover this important intra-day stylised fact of financial markets.

5 Conclusion

To date, no successful calibrations of agent based models to market micro-structure have been observed and literature has focused on stylised fact-centric model validation rather than empirically optimising for agent based model parameters (Platt and Gebbie, 2018). We attempted to calibrate the PGPS Model (Preis et al., 2006) to high frequency trading time series data using SMC ABC as a calibration technique to statistically optimise the parameters and output Bayesian credibility intervals to measure the degree of precision attained. To do this we reconstructed the PGPS Model (Preis et al., 2006) in Python, ensured that the model is implemented correctly by replicating and validating the model results produced by Platt and Gebbie (2018). We then validated that the SMC ABC calibration is implemented correctly by testing it on a simple autoregressive moving average model. Using this SMC ABC implementation we calibrated the PGPS Model (Preis et al., 2006) using synthetic data where the underlying parameters are known and find the calibration to be mostly unsuccessful for this complex model even after incorporating more summary statistics. We then calibrate the PGPS Model (Preis et al., 2006) using real-world JSE trading data and find that the calibrated model is unable to correctly recover stylised facts.

We find that, although SMC ABC's naive approach of updating distributions can successfully calibrate simple toy models, such as autoregressive moving average models, it fails to calibrate this agent based model for high frequency trading. Furthermore, the calibrated model fails to recover key stylised facts, namely the lack of auto-correlation of returns and order-flow clustering and tends to exaggerate both the leptokurtosis of returns and volatility clustering relative to the real-world high frequency data to which it is calibrated. However, we have found that SMC ABC is useful for visualising successive posterior distributions and whether they iteratively converge to a true underlying posterior distribution. This enables the user of the SMC ABC algorithm to identify whether the calibration was successful or not in identifying a uniquely identifiable set of parameters. The Bayesian nature of SMC ABC then allows for a more intuitive

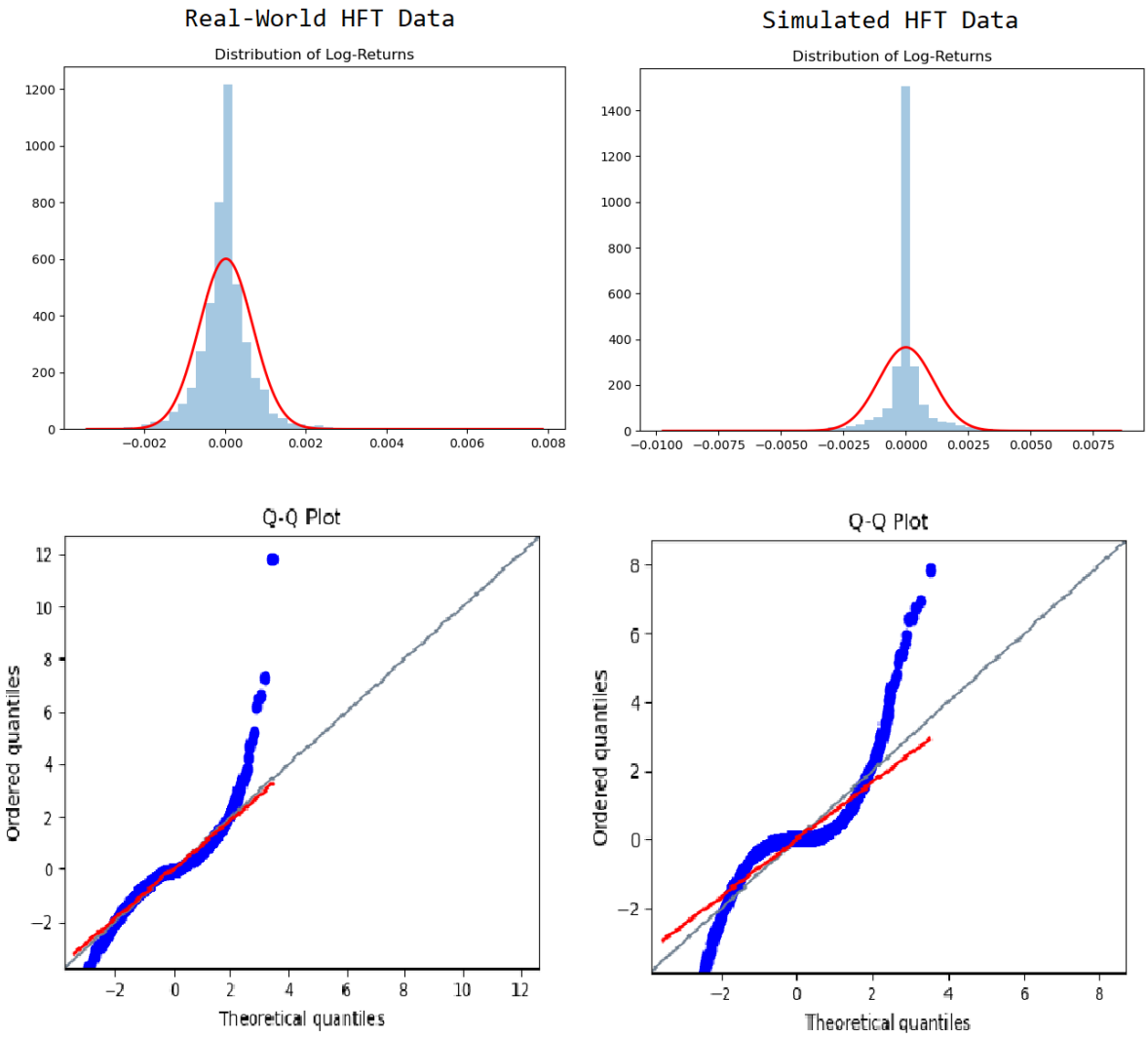

Figure 14: Plots comparing the distribution of log-returns for the real-world high frequency data and the simulated high frequency data where the simulated data is obtained by simulating the PGPS Model (Preis et al., 2006) using the parameters calibrated to Anglo American PLC trading data. Leptokurtic returns are evident in both data sets, as evident by the heavier tails in the QQ-plot with respect to the Gaussian distribution and high peaks in the plotted distribution as compared to Gaussian. However, the simulated data has exaggerated leptokurtic returns in comparison to the real-world returns to which it was calibrated.

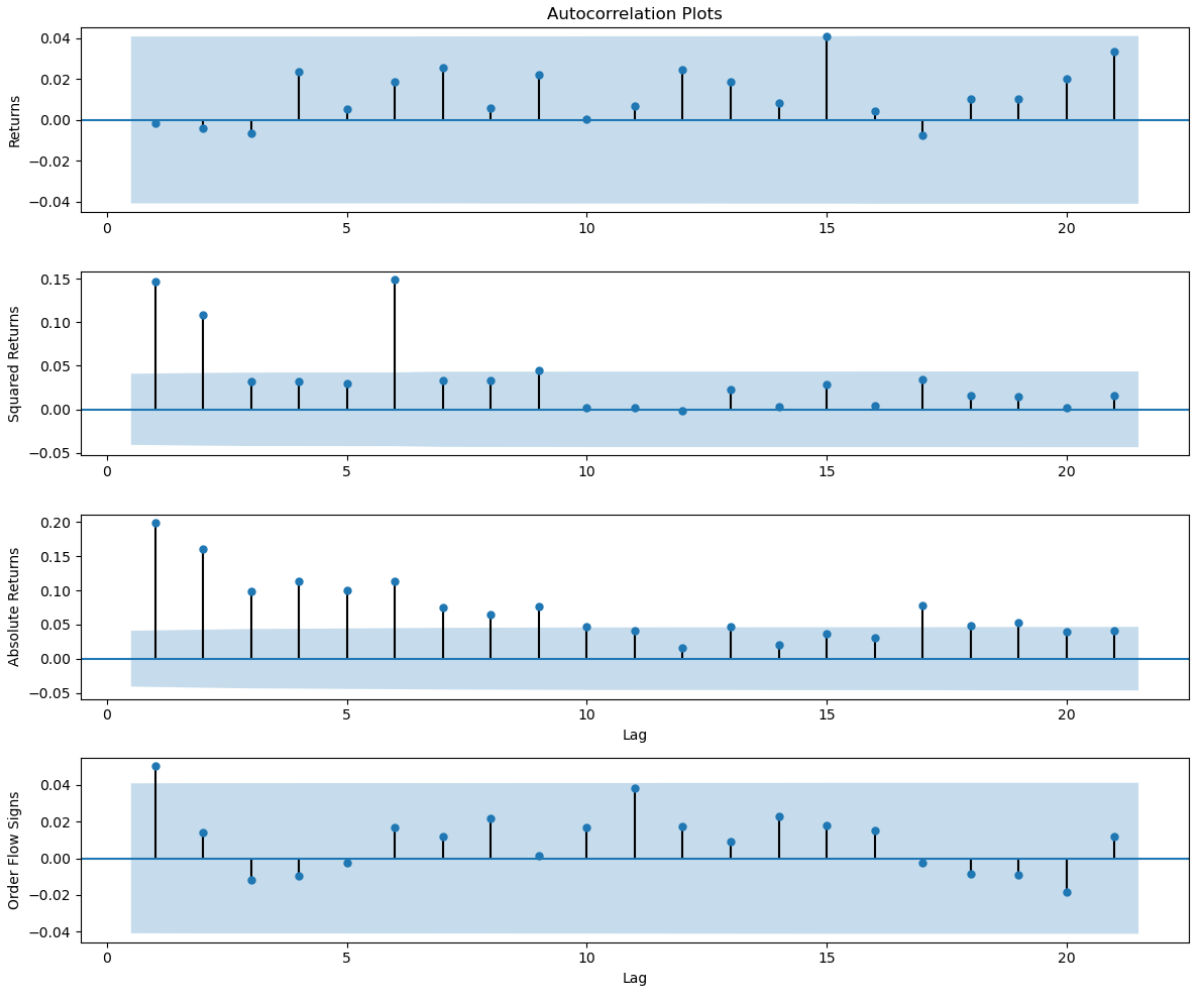

Figure 15: Plots illustrating the auto-correlations of log-returns, squared log-returns, absolute log-returns and order-flow signs for Anglo American PLC with 95% confidence interval bands where the standard deviation is computed according to Bartlett's formula (Bartlett, 1946). Evidently there is absence of auto-correlations of raw log-returns while the auto-correlations of squared and absolute returns are positive and slowly decay illustrating volatility clustering. The auto-correlations of trade signs are initially positive which is indicative of order-flow clustering.

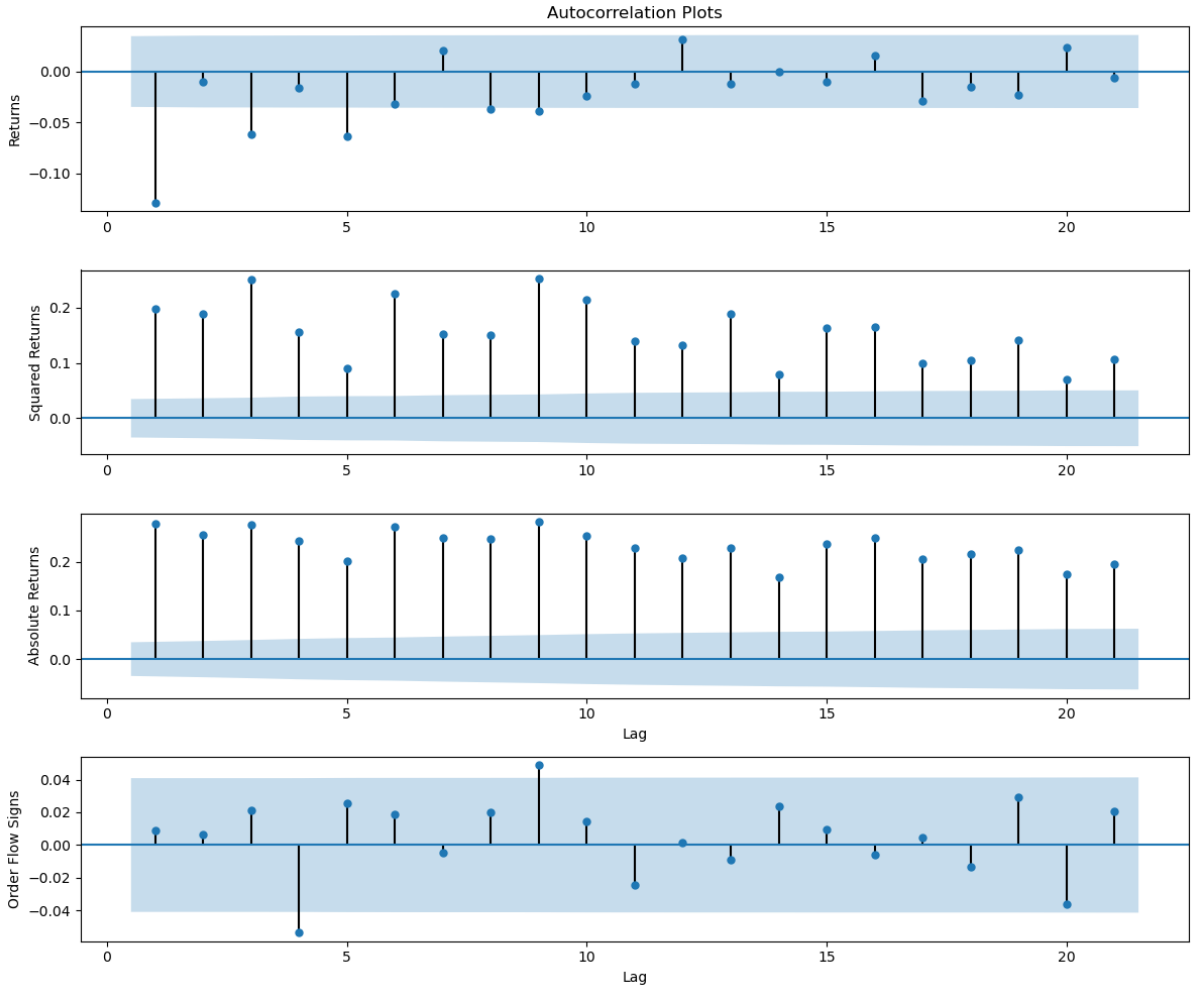

Figure 16: Plots illustrating the auto-correlations of log-returns, squared log-returns, absolute log-returns and order-flow signs for the simulated trading data with 95% confidence interval bands where the standard deviation is computed according to Bartlett's formula (Bartlett, 1946). Evidently, auto-correlations of raw log-returns are negative and do not decay quickly to zero implying that the stylised fact of absence of auto-correlation of raw returns does not hold for the simulated data. However, volatility clustering is evident by the positive auto-correlations of squared and absolute returns but this seems exaggerated in magnitude as well as showing particularly slow decay to zero in comparison to Anglo American PLC data to which it was calibrated. Order-flow clustering is not evident as auto-correlations of 1-3 lags are not significantly positive and the 4th lag is significantly negative.

assessment of the posterior distribution and its posterior means, modes and medians rather than a single point estimate given by a Frequentist approach.

The failure of the empirical calibration may be for two key reasons, either the parameters of the model are not uniquely identifiable given the model output or the SMC ABC rejection mechanism results in information loss rendering parameters unidentifiable given the insufficient summary statistics. Parameter identifiability is a known issue particularly for models with many parameters and relative complexity and the PGPS Model (Preis et al., 2006) is one such case (Platt, 2020). This may be a reason why this particular model has never been successfully calibrated and why similarly complex agent based models for noisy market micro-structure data have never been successfully empirically calibrated (Platt and Gebbie, 2018). Additionally, while ABC calibrations, being Bayesian in nature, are effective in providing posterior distributions and credibility intervals, there is information loss when using these rejection algorithms on data aggregated into summary statistics rather than maximizing a simulated likelihood directly. Platt (2020) has found in his paper "A comparison of economic agent based model calibration methods" that although Bayesian methods consistently outperform Frequentist methods, the construction and optimisation of a likelihood is a key component of success across calibration techniques in general, independently of whether the approach is Frequentist or Bayesian. This is due to lack of information loss associated with likelihood approaches as opposed to distance rejection approaches which require summary statistics for efficiency limitations. The efficiency constraint which requires the use of summary statistics means that SMC ABC is unlikely to be successful for larger more complex models with large parameter spaces, because even if the parameters had a weak effect on the model output and are mildly collinear but still inherently uniquely identifiable, summarising the output may reduce the information such that one or more of the parameters is no longer uniquely identifiable. This is a particular limitation of SMC ABC algorithms for calibrating complex algorithms, because larger more complex models tend to have greater computational complexity and, therefore, require more succinctly aggregated summary statistics but simultaneously require less information loss to ensure the large number of parameters are still uniquely identifiable.

We found that overall the success of the calibration of complex models is heavily constrained by the computational complexity of both the underlying model and the implementation of the calibration technique. As such, profiling and minimising model simulation time and the use of high performance clustering or cloud computing are essential prerequisites for calibrating complex agent based models regardless of the calibration technique employed.

However, it is evident that likelihood approaches used in a Bayesian context, such as the calibration method introduced by Grazzini et al. (2017), are more likely to be successful for complex agent based models for noisy market micro-structure time series data (Platt, 2020). Subsequently, to extend this research we recommend implementing Bayesian likelihood based calibration techniques on agent based models for high frequency trading data.

References

Y. Aït-Sahalia, P. A. Mykland, and L. Zhang. Ultra high frequency volatility estimation with dependent microstructure noise. *Journal of Econometrics*, 160(1):160–175, 2011.

S. Barde. Direct comparison of agent-based models of herding in financial markets. *Journal of Economic Dynamics and Control*, 73:329–353, 2016.

M. S. Bartlett. On the theoretical specification and sampling properties of autocorrelated time-series. *Supplement to the Journal of the Royal Statistical Society*, 8(1):27–41, 1946.

M. A. Beaumont, J.-M. Cornuet, J.-M. Marin, and C. P. Robert. Adaptive approximate bayesian computation. *Biometrika*, 96(4):983–990, Dec 2009. doi: 10.1093/biomet/asp052.

B. Biais, P. Hillion, and C. Spatt. An empirical analysis of the limit order book and the order flow in the paris bourse. *the Journal of Finance*, 50(5):1655–1689, 1995.

F. Black and M. Scholes. The pricing of options and corporate liabilities. *Journal of political economy*, 81 (3):637–654, 1973.

J. R. Blevins. Sequential monte carlo methods for estimating dynamic microeconomic models. *Journal of Applied Econometrics*, 31(5):773–804, 2016.

R. Bookstaber, M. Paddrik, and B. Tivnan. An agent-based model for financial vulnerability. *Journal of Economic Interaction and Coordination*, 13(2):433–466, 2018.

P. Bortot, S. G. Coles, and S. A. Sisson. Inference for stereological extremes. *Journal of the American Statistical Association*, 102(477):84–92, 2007. doi: 10.1198/016214506000000988.

J.-P. Bouchaud, Y. Gefen, M. Potters, and M. Wyart. Fluctuations and response in financial markets: the subtle nature of 'random'price changes. *Quantitative finance*, 4(2):176–190, 2004.

W. A. Brock and C. H. Hommes. Heterogeneous beliefs and routes to chaos in a simple asset pricing model. *Journal of Economic Dynamics and Control*, 22(8-9):1235–1274, 1998. doi: 10.1016/s0165-1889(98) 00011-6.

S. Bruzzone and M. Kaveh. Information tradeoffs in using the sample autocorrelation function in arma parameter estimation. *IEEE transactions on acoustics, speech, and signal processing*, 32(4):701–715, 1984.

A. Butler and C. Glasbey. Corrigendum: A latent gaussian model for compositional data with zeros. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 58(1):141–141, 2009. doi: 10. 1111/j.1467-9876.2008.00644.x.

A. Chakraborti, I. M. Toke, M. Patriarca, and F. Abergel. Econophysics review: I. empirical facts. *Quantitative Finance*, 11(7):991–1012, 2011.

C. Chiarella and G. Iori. A simulation analysis of the microstructure of double auction markets*. *Quantitative Finance*, 2(5):346–353, Jan 2002. doi: 10.1088/1469-7688/2/5/303.

C. Chiarella, G. Iori, and J. Perelló. The impact of heterogeneous trading rules on the limit order book and order flows. *Journal of Economic Dynamics and Control*, 33(3):525–537, 2009. doi: 10.1016/j.jedc. 2008.08.001.

R. Cont. Empirical properties of asset returns: stylized facts and statistical issues. 2001.

G. Fagiolo. Macroeconomic policy in dsge and agent-based models redux: New developments and challenges ahead. *SSRN Electronic Journal*, 2016. doi: 10.2139/ssrn.2763735.

J. Fan, C. Zhang, and J. Zhang. Generalized likelihood ratio statistics and wilks phenomenon. *Annals of statistics*, pages 153–193, 2001.

J. Farmer and S. Joshi. The price dynamics of common trading strategies. *Journal of Economic Behavior Organization*, 49(2):149–171, 2002. doi: 10.1016/s0167-2681(02)00065-3.

J. D. Farmer and J. Geanakoplos. The virtues and vices of equilibrium and the future of financial economics. *Complexity*, 14(3):11–38, 2009.

J. D. Farmer, P. Patelli, and I. I. Zovko. The predictive power of zero intelligence in financial markets. *Proceedings of the National Academy of Sciences*, 102(6):2254–2259, 2005.

P. Fearnhead and D. Prangle. Constructing summary statistics for approximate bayesian computation: semi-automatic approximate bayesian computation. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 74(3):419–474, 2012. doi: 10.1111/j.1467-9868.2011.01010.x.

S. Filippi, C. P. Barnes, J. Cornebise, and M. P. Stumpf. On optimality of kernels for approximate bayesian computation using sequential monte carlo. *Statistical applications in genetics and molecular biology*, 12 (1):87–107, 2013.

A. R. Gallant, H. Hong, and A. Khwaja. A bayesian approach to estimation of dynamic models with small and large number of heterogeneous players and latent serially correlated states. *Journal of Econometrics*, 203(1):19–32, 2018.

T. Gebbie and D. Platt. Aglj.j 5 days of 1-minute mid-price bar test data, 2019. URL https://data.mendeley.com/datasets/nt8nw28h7c/1.

D. K. Gode and S. Sunder. Allocative efficiency of markets with zero-intelligence traders: Market as a partial substitute for individual rationality. *Journal of political economy*, 101(1):119–137, 1993.

K. Goosen and T. Gebbie. High frequency trading agent based model calibration using sequential monte carlo approximate bayesian computation : Software, Sep 2020. URL https://zivahub.uct.ac.za/articles/software/High_frequency_trading_agent_based_model_calibration_using_sequential_Monte_Carlo_approximate_Bayesian_Computation_Software/12894005/1.

M. S. Granero, J. T. Segovia, and J. G. Pérez. Some comments on hurst exponent and the long memory processes on capital markets. *Physica A: Statistical Mechanics and its applications*, 387(22):5543–5551, 2008.

J. Grazzini and M. Richiardi. Estimation of ergodic agent-based models by simulated minimum distance. *Journal of Economic Dynamics and Control*, 51:148–165, 2015.

J. Grazzini, M. G. Richiardi, and M. Tsionas. Bayesian estimation of agent-based models. *Journal of Economic Dynamics and Control*, 77:26–47, 2017.

M. Guerini and A. Moneta. A method for agent-based models validation. *Journal of Economic Dynamics and Control*, 82:125–141, 2017.

J. Hodges. The significance probability of the smirnov two-sample test. *Arkiv för Matematik*, 3(5):469–486, 1958.

A. Imle, P. Kumberger, N. D. Schnellbächer, J. Fehr, P. Carrillo-Bustamante, J. Ales, P. Schmidt, C. Ritter, W. J. Godinez, B. Müller, et al. Experimental and computational analyses reveal that environmental restrictions shape hiv-1 spread in 3d cultures. *Nature communications*, 10(1):1–18, 2019.

M. C. Jensen. Some anomalous evidence regarding market efficiency. *Journal of Financial Economics*, 6 (2-3):95–101, 1978. doi: 10.1016/0304-405x(78)90025-9.

S. H. John W. Eaton, David Bateman and R. Wehbring. *GNU Octave version 3.8.1 manual: a high-level interactive language for numerical computations*. CreateSpace Independent Publishing Platform, 2014. URL http://www.gnu.org/software/octave/doc/interpreter. ISBN 1441413006.

P. Joyce and P. Marjoram. Approximately sufficient statistics and bayesian computation. *Statistical applications in genetics and molecular biology*, 7(1), 2008.

N. Kambhatla and T. K. Leen. Dimension reduction by local principal component analysis. *Neural computation*, 9(7):1493–1516, 1997.

P. Klimek, S. Poledna, J. D. Farmer, and S. Thurner. To bail-out or to bail-in? answers from an agent-based model. *Journal of Economic Dynamics and Control*, 50:144–154, 2015.

E. Klinger, D. Rickert, and J. Hasenauer. pyabc: distributed, likelihood-free inference. *Bioinformatics*, 34 (20):3591–3593, 2018.

E. G. Klinger. *Approximate Bayesian Model Selection for Local Cortical Networks at Cellular Resolution*. PhD thesis, Technische Universität München, 2018.

F. Lamperti. An information theoretic criterion for empirical validation of time series models. *SSRN Electronic Journal*, 2015. doi: 10.2139/ssrn.2570828.

F. Lamperti and A. Sani. Agent-based model calibration using machine learning surrogates. *SSRN Electronic Journal*, 2017. doi: 10.2139/ssrn.2943297.

B. LeBaron. Empirical regularities from interacting long-and short-memory investors in an agent-based stock market. *Ieee transactions on evolutionary computation*, 5(5):442–455, 2001.

B. LeBaron. Agent-based computational finance. *Handbook of computational economics*, 2:1187–1233, 2006.

T. Lux. Estimation of agent-based models using sequential monte carlo methods. *Journal of Economic Dynamics and Control*, 91:391–408, 2018. doi: 10.1016/j.jedc.2018.01.021.

B. G. Malkiel. The efficient market hypothesis and its critics. *Journal of economic perspectives*, 17(1): 59–82, 2003.

B. Mandelbrot. The variation of some other speculative prices. *The Journal of Business*, 40(4):393–413, 1967.

P. Marjoram and S. Tavaré. Modern computational approaches for analysing molecular genetic variation data. *Nature Reviews Genetics*, 7(10):759–770, 2006. doi: 10.1038/nrg1961.

S. Maslov. Simple model of a limit order-driven market. *Physica A: Statistical Mechanics and its Applications*, 278(3-4):571–578, 2000.

MATLAB. *version 7.10.0 (R2010a)*. The MathWorks Inc., Natick, Massachusetts, 2010.

M. Matsumoto and T. Nishimura. Mersenne twister: a 623-dimensionally equidistributed uniform pseudo-random number generator. *ACM Transactions on Modeling and Computer Simulation (TOMACS)*, 8 (1):3–30, 1998.

F. Mcgroarty, A. Booth, E. Gerding, and V. L. R. Chinthalapati. High frequency trading strategies, market fragility and price spikes: an agent based model perspective. *Annals of Operations Research*, 282(1-2): 217–244, 2018. doi: 10.1007/s10479-018-3019-4.

F. McGroarty, A. Booth, E. Gerding, and V. R. Chinthalapati. High frequency trading strategies, market fragility and price spikes: an agent based model perspective. *Annals of Operations Research*, 282(1-2): 217–244, 2019.

P. D. Moral, A. Doucet, and A. Jasra. An adaptive sequential monte carlo method for approximate bayesian computation. *Statistics and Computing*, 22(5):1009–1020, Mar 2011. doi: 10.1007/s11222-011-9271-y.

M. A. Nunes and D. J. Balding. On optimal selection of summary statistics for approximate bayesian computation. *Statistical applications in genetics and molecular biology*, 9(1), 2010.

E. Panayi, M. Harman, and A. Wetherilt. Agent-based modelling of stock markets using existing order book data. In *International Workshop on Multi-Agent Systems and Agent-Based Simulation*, pages 101–114. Springer, 2012.

T. Peters, B. Meyer, R. Stuike-Prill, R. Somorjai, and J.-R. Brisson. A monte carlo method for conformational analysis of saccharides. *Carbohydrate research*, 238:49–73, 1993.

D. Platt. A comparison of economic agent-based model calibration methods. *Journal of Economic Dynamics and Control*, 113:103859, 2020.

D. Platt and T. Gebbie. Can agent-based models probe market microstructure? *Physica A: Statistical Mechanics and its Applications*, 503:1092–1106, 2018. doi: 10.1016/j.physa.2018.08.055.

V. Plerou and H. E. Stanley. Stock return distributions: tests of scaling and universality from three distinct stock markets. *Physical Review E*, 77(3):037101, 2008.

D. Prangle et al. Adapting the abc distance function. *Bayesian Analysis*, 12(1):289–309, 2017.

T. Preis, S. Golke, W. Paul, and J. J. Schneider. Multi-agent-based order book model of financial markets. *Europhysics Letters (EPL)*, 75(3):510–516, 2006. doi: 10.1209/epl/i2006-10139-0.

R. Roll and S. A. Ross. An empirical investigation of the arbitrage pricing theory. *The Journal of Finance*, 35(5):1073–1103, 1980.

S. A. Sisson, Y. Fan, and M. M. Tanaka. Sequential monte carlo without likelihoods. *Proceedings of the National Academy of Sciences*, 104(6):1760–1765, 2007. ISSN 0027-8424. doi: 10.1073/pnas.0607208104. URL https://www.pnas.org/content/104/6/1760.

J. Staccioli and M. Napoletano. An agent-based model of intra-day financial markets dynamics. *Journal of Economic Behavior & Organization*, 2020.

M. M. Tanaka, A. R. Francis, F. Luciani, and S. A. Sisson. Using approximate bayesian computation to estimate tuberculosis transmission parameters from genotype data. *Genetics*, 173(3):1511–1520, 2006. doi: 10.1534/genetics.106.055574.

Thomsonreuters.com. Tick history. *URL: https://tickhistory.thomsonreuters.com [accessed 02 February, 2019]*, 2019.

T. Toni, D. Welch, N. Strelkowa, A. Ipsen, and M. P. Stumpf. Approximate bayesian computation scheme for parameter inference and model selection in dynamical systems. *Journal of the Royal Society Interface*, 6(31):187–202, 2009.

R. S. Tsay. *Analysis of financial time series*, volume 543. John wiley & sons, 2005.

G. Van Rossum and F. L. Drake. *Python 3 Reference Manual*. CreateSpace, Scotts Valley, CA, 2009. ISBN 1441412697.

R. D. Wilkinson. Approximate bayesian computation (abc) gives exact results under the assumption of model error. *Statistical applications in genetics and molecular biology*, 12(2):129–141, 2013.
.

6 Appendices

A GitHub Repository: hft-abm-smc-abc (Goosen and Gebbie, 2020)

A.1 Setup requirements

Install poetry from here https://python-poetry.org/docs/installation. Then, after cloning the hft-abm-smc-abc repository, in the terminal run 'poetry install'. This automatically installs all the libraries required to run the code in the hft-abm-smc-abc repository (Goosen and Gebbie, 2020).

A.2 The PGPS Model Code

To run the PGPS Model (Preis et al., 2006) the Python script files preisOrderBookSeed.py and preisSeed.py along with the config.py are required to initialise the limit order book and execute PGPS Model (Preis et al., 2006) given the parameters specified in the config.py file.

A.3 SMC ABC using pyABC

To calibrate the PGPS Model (Preis et al., 2006) using pyABC to implement SMC ABC simply run the Python script `SMC_ABC.py`. This uses the SMC ABC configurations supplied in `SMC_ABC_init.py`. Note that this requires a Linux operating system to run since the parallelisation scheme employed in pyABC is not setup for Windows. To run the calibration on the University of Cape Town's HPC submit the batch script `test_parallel.sh`.

A.4 Visualising Outputs

To extract the SMC ABC results and visualise them, refer to the Python script openSMCABChistory.py. Additionally, to plot the stylised facts of financial time series make use of the script `stylised_facts.py`.

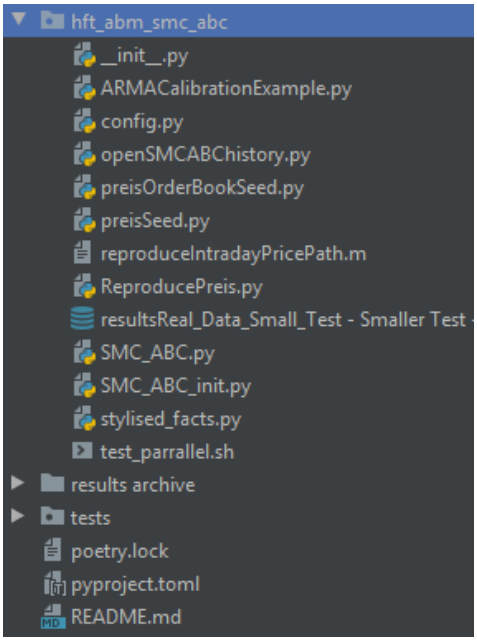

Figure 17: Screenshot showing the repository containing the script files used to reproduce the results in this research (Goosen and Gebbie, 2020).

B Diagrams

B.1 Preis-Golke-Paul-Schneider Agent Based Model (Preis et al., 2006) Flow Diagram

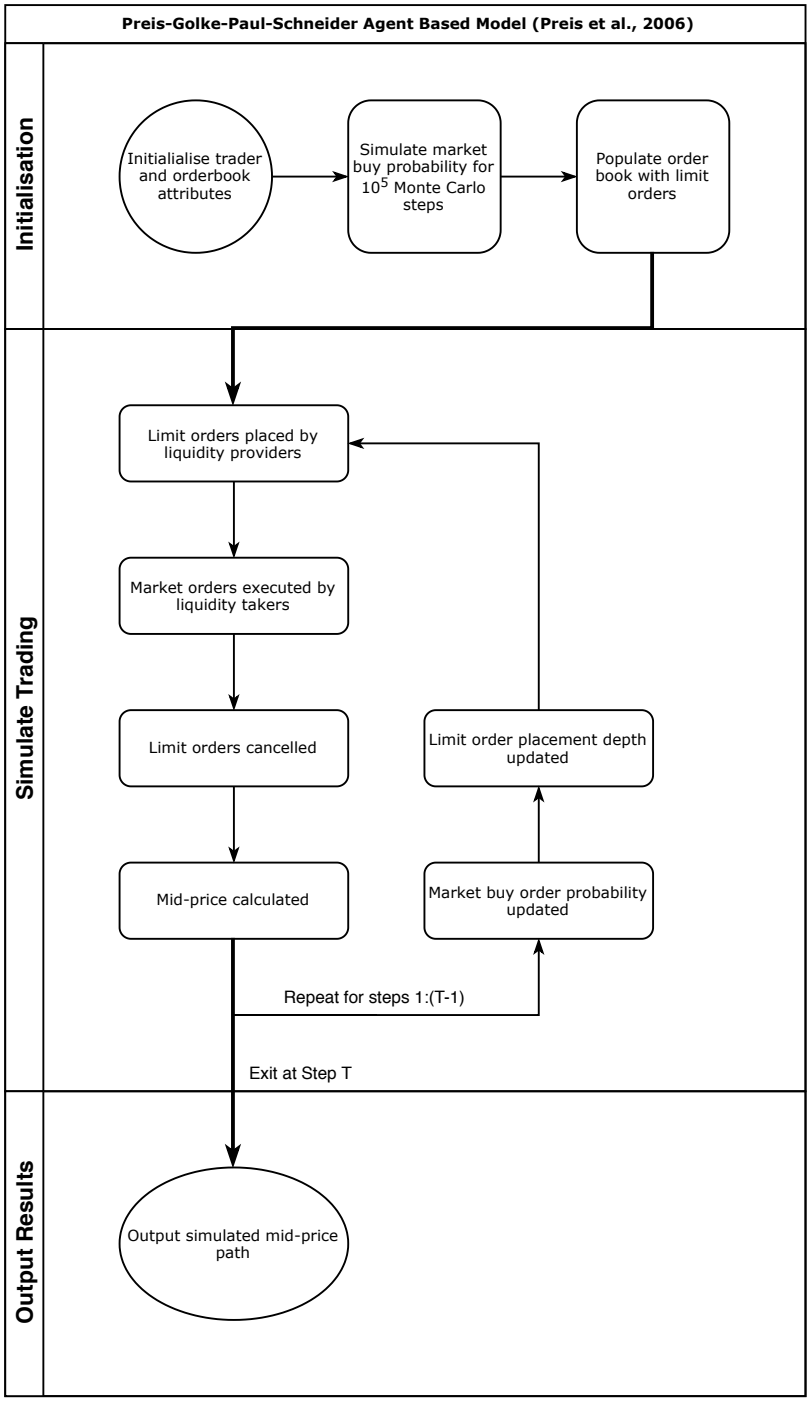

Figure 18: Flow diagram illustrating the initialisation and simulation of the PGPS Model (Preis et al., 2006) for T Monte Carlo Steps.

B.2 SMC ABC Flow Chart

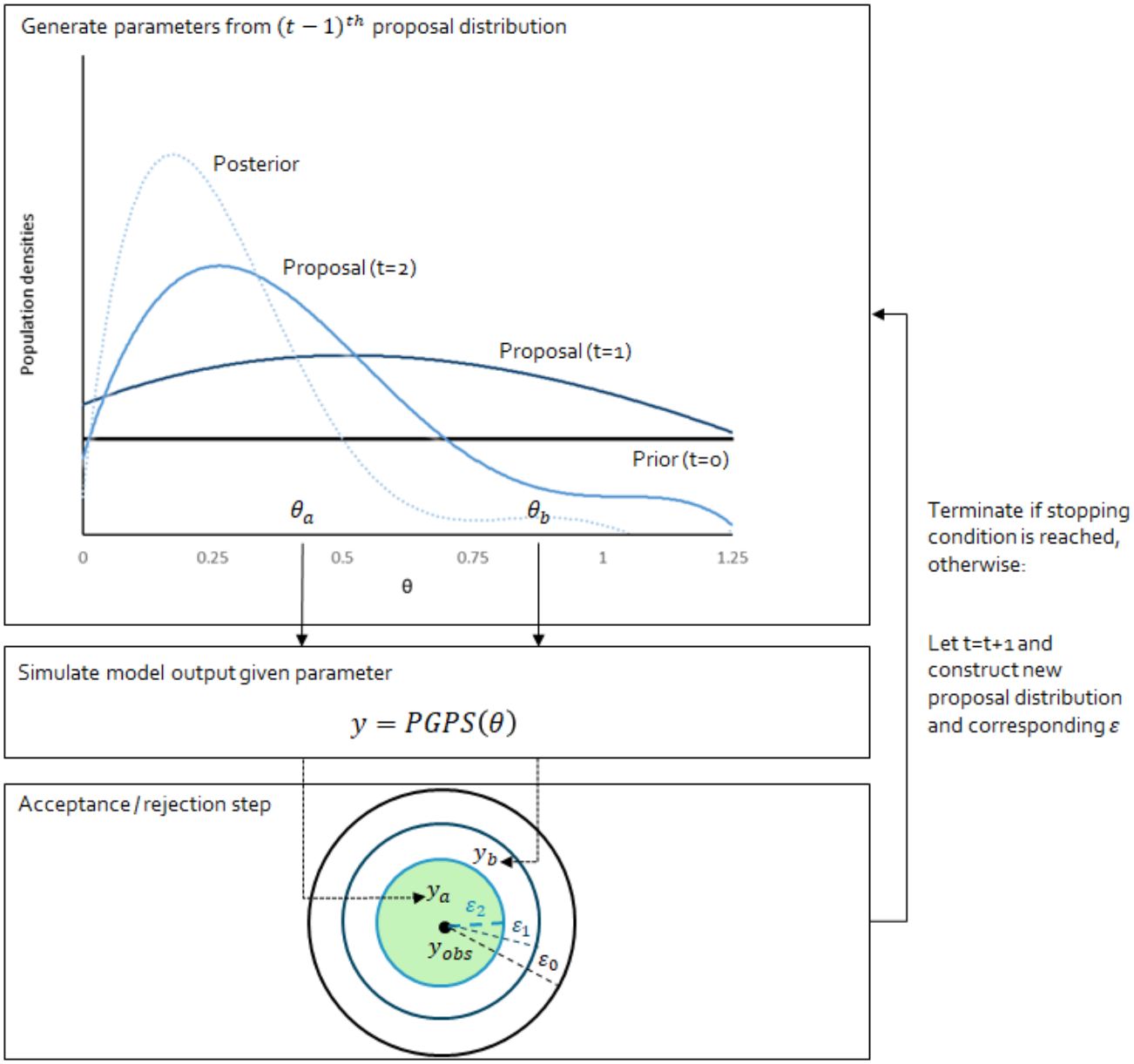

Figure 19: Flow chart illustrating an overview of the SMC ABC calibration algorithm for the PGPS Model (Preis et al., 2006). Initially, parameters are sampled from the flat uniform prior at $t = 0$ with corresponding user specified acceptance threshold value, ϵ_0. At each subsequent iteration, t, a new proposal distribution is determined and converges to the true posterior distribution. Each subsequent proposal distribution is used to generate the parameters. Additionally, with each subsequent t, the threshold acceptance value, ϵ_t, decreases. In this figure two proposal parameters are generated from the proposal distribution at $t = 2$. These two proposals are then used in PGPS Model (Preis et al., 2006) to simulate two sets of output, y_a and y_b, given these parameters. Proposal θ_a is accepted as the simulated output, y_a, is less than the ϵ_2 acceptance threshold away from the observed value, y_{obs}. Whereas, proposal θ_b is rejected because the observations, y_{obs}, and the simulated output, y_b, given this proposal are too dissimilar (as measured by the acceptance threshold, ϵ_2). Note that at $t = 1$ proposal θ_2 would have been accepted. The accuracy increases with each subsequent population of the SMC ABC algorithm as ϵ_t decreases.

C Algorithms

C.1 Simple ABC

The simple ABC algorithm samples from the posterior distribution, $p(\theta \mid y_{obs})$ without needing an explicit expression for the likelihood function.

Algorithm 4 Simple ABC

Require:

1. Sample candidate parameters from prior, $\theta^* \sim \pi(\theta)$.
2. Given candidate parameter, θ^*, simulate $s^* = S(f(y^*|\theta^*))$.
3. Reject with probability proportional to $K\left(d(s^*, s_{obs})\right)$.
4. Repeat Steps 1-3 until a sufficiently large sample is obtained.

C.2 Markov Chain Monte Carlo (MCMC) ABC

The algorithm alters the acceptance rejection step given in the simple ABC algorithm (Algorithm C.1) by instead using a likelihood ratio statistic and altering the proposal distribution and thereby follows a typical Metropolis Hasting algorithm.

Algorithm 5 MCMC ABC

Require:

1. At $t = 1$:
   1.1. Select ϵ.
   1.2. Sample $\theta^{(1)}$ (i.e. from $\theta^{(1)} \sim \pi(\theta)$).
2. Propose θ' from a Metropolis Hastings kernel (i.e. from $\theta' \sim q(\theta' \mid \theta^{(t)})$).
3. Simulate $s(y) \sim s(f(y \mid \theta'))$.
4. Let
$$\theta^{(t+1)} = \begin{cases} \theta' & \text{w.p. } \min\left(1, \frac{\pi(\theta')q(\theta^{(t)}|\theta')}{\pi(\theta^{(t)})q(\theta'|\theta^{(t)})} I(d(S(y_{obs}), S(y)) \leq \epsilon)\right) \\ \theta^{(t)} & otherwise \end{cases} \tag{15}$$
5. Increment $t = t + 1$.
6. Repeat steps 2-5 until convergence.

C.3 Population Monte Carlo (PMC) ABC

Algorithm 6 PMC ABC

Require: Repeat for t=1,2,... until termination is condition reached.

1. Repeat Steps 1-4 of the Simple ABC Algorithm C.1, except for $t > 1$ let $\pi(\theta) = q_t(\theta)$, until B particles are obtained, where:
$$q_t(\theta) = \left(\sum_{i=1}^{B} w_i^{t-1} K_t^p(\theta \mid \theta_i^{t-1})\right) / \sum_{i=1}^{B} w_i^{t-1}. \tag{16}$$
2. Let the importance weight $w_i^t = \pi(\theta_i^t) q_t(\theta_i^t)$.

$K_t^p(\theta \mid \theta')$ is the proposal kernel and is often taken to be $K_t^p(\theta \mid \theta') = N(\theta', 2\Sigma_{t-1})$ and Σ_{t-1} is the empirical covariance matrix of the particles computed at iteration $t-1$.

D Proofs

D.1 PGPS Model (Preis et al., 2006) Parameter Space Constraints

Let $N(t)$ be the number of limit orders in the order book at time t and N_A the number of limit orders and number of market orders, then, recursively, using the order rates α, μ and δ we know the number of limit orders at time $t+1$ is given by:

$$N(t+1) = N(t) + \alpha N_A - \delta(N(t) + \alpha N_A) - \mu N_A. \tag{17}$$

Hence, in equilibrium:

$$\frac{N_{\text{equilibrium}}}{N_A} = \frac{\alpha(1-\delta)}{\delta} - \frac{\mu}{\delta}. \tag{18}$$

Subsequently, the order book is stable if and only if the following conditions hold:

$$\delta > 0, \text{ and } \alpha(1-\delta) > \mu.$$